

Article

Crossing-Point Estimation in Human–Robot Navigation—Statistical Linearization versus Sigma-Point Transformation

Rainer Palm ^{1,*} and Achim J. Lilienthal ²

¹ Center for Applied Autonomous Sensor Systems (AASS), Department of Technology, Örebro University, SE-701 82 Örebro, Sweden

² Technical University Munich (TUM), 80333 Munich, Germany; achim.j.lilienthal@tum.de

* Correspondence: rub.palm@t-online.de

Abstract: Interactions between mobile robots and human operators in common areas require a high level of safety, especially in terms of trajectory planning, obstacle avoidance and mutual cooperation. In this connection, the crossings of planned trajectories and their uncertainty based on model fluctuations, system noise and sensor noise play an outstanding role. This paper discusses the calculation of the expected areas of interactions during human–robot navigation with respect to fuzzy and noisy information. The expected crossing points of the possible trajectories are nonlinearly associated with the positions and orientations of the robots and humans. The nonlinear transformation of a noisy system input, such as the directions of the motion of humans and robots, to a system output, the expected area of intersection of their trajectories, is performed by two methods: statistical linearization and the sigma-point transformation. For both approaches, fuzzy approximations are presented and the inverse problem is discussed where the input distribution parameters are computed from the given output distribution parameters.

Keywords: human–robot interaction; Gaussian noise; sigma-point transformation; unscented Kalman filter



Citation: Palm, R.; Lilienthal, A.J.

Crossing-Point Estimation in Human–Robot Navigation—Statistical Linearization versus Sigma-Point Transformation. *Sensors* **2024**, *24*, 3303. <https://doi.org/10.3390/s24113303>

Academic Editors: David Cheneler and Stephen Monk

Received: 20 March 2024

Revised: 16 May 2024

Accepted: 17 May 2024

Published: 22 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The planning and performing of mobile robot tasks in the presence of human operators while sharing the same workspace requires a high level of stability and safety. Research activities regarding navigation, obstacle avoidance, adaptation and collaboration between robots and human agents have been widely reported [1,2]. Multiple target tracking for robots using higher control levels in a control hierarchy are discussed in [3,4]. A human-friendly interaction between robots and humans can be obtained by human-like sensor systems [5]. A prominent role in robot navigation is the trajectory-crossing problem of robots and humans [6,7] and corresponding fuzzy solutions [8]. Motivations for a fuzzy solution of the intersection problem are manifold. One point is an uncertain measurement of the position and orientation of the human agent, because of which the use of a fuzzy signal and an adequate fuzzy processing seems natural [9,10]. Another aspect is the need for decreasing the computing effort in the case of complex calculations during a very small time interval. System uncertainties and observation noise lead to uncertainties of the intersection estimations.

The objective of this work is the formulation of the crossing/intersection problem by taking into account the uncertainties in human–robot systems, including sensors and motor characteristics. An important aspect is to define permissible uncertainties in a human–robot system for a given uncertainty at a possible intersection of their trajectories. Taking into account the nonlinearities, this is performed by the differential approach and a following analysis of the regarding Gaussian distributions. This approach is compared with the

sigma-point transformation, which represents a simplification of the computation and a qualitative extension of the analysis regarding the statistics of the random variables. For broader areas of possible intersections, both methods are extended to fuzzy regions together with different numbers and shapes of fuzzy sets. The most important contributions are as follows:

- An investigation of uncertainties of possible intersection areas originating from sensor noise or system uncertainties.
- A direct and inverse transformation of the error variables at the intersection areas for two input variables (orientation angles) and two output variables (intersection coordinates).
- An extension of the method from two to six input variables (two orientation angles and four position coordinates).
- An exploration of the formulations of fuzzy versions.
- A formulation of the problem by the sigma-point transformation and corresponding comparison of the two methods.

This paper deals with the one-robot one-human trajectory-crossing problem, where small uncertainties in the position and orientation may lead to high uncertainties at the intersection points. The position and orientation of the human and robot are nonlinearly coupled but can be linearized. In the following, the linear part of the nonlinear system is considered in the analysis reported for small variations in the input [11]. Then, the “direct task” is described, meaning that the parameters of the input distribution are transformed to the output distribution parameters. The “inverse task” is also solved, meaning that for the defined output distribution parameters the input parameters are calculated. In this paper, two methods are outlined:

1. The *statistical linearization*, which linearizes the nonlinearity around the operating area at the intersection. The means and standard deviations on the input parameters positions (orientations) are transformed through the linearized nonlinear system to obtain the means and standard deviations of the output parameters (the position of intersection).
2. The *sigma-point transformation*, which calculates the so-called sigma points of the input distribution, including the mean and covariance of the input. The sigma points are directly propagated through the nonlinear system [12–14] to obtain the means and covariance of the output and, with this, the standard deviations of the output (the position of intersection). The advantage of the sigma-point transformation is that it captures the first- and second-order statistics of a random variable, whereas the statistical linearization approximates a random variable only by its first order. However, the computational complexity of the extended Kalman filter (EKF, differential approach) and unscented Kalman filter (UKF, sigma-point approach) is of the same order [13].

This paper is organized as follows. Section 2 describes the related work already conducted on unscented Kalman filters in mobile robot applications. In Section 3, the general intersection problem and its analytical approach is described. Section 4 deals with the transformation/conversion of Gaussian distributions for a two-input–two-output system and for a six-input–two-output system plus the corresponding inverse and fuzzy solutions. In Section 5, the sigma-point approach plus inverse and fuzzy solutions are addressed. Section 6 presents simulations of the *statistical linearization* and the *sigma-point transformation* to show the quality of the input–output conversion of the distributions and the impact of different resolutions of fuzzy approximations on the accuracy of the random variable intersection. Finally, Section 7 concludes this paper with a discussion of the two different approaches and a comparison of the methods.

2. Related Work

The crossing problem for mobile robots has been especially dealt with by [6,7]. Both publications deal with the so-called rendezvous problem whereby the key point is the trajectory planning under time constraints, taking into account the dynamics of the con-

tributing robots. Uncertainties of possible intersection areas that come from sensor noise or system uncertainties are not discussed deeply. A fuzzy-adaptive extended Kalman filter (FAEKF) for the real-time attitude estimation of a mobile robot is proposed in [15] where fuzzy IF–THEN rules-based adaption laws modify the noise covariance matrices of the filter. However, the use of unscented Kalman filters or sigma-point transformation has not been discussed. For the estimation of landmarks, a simultaneous localization and mapping (SLAM) method is presented by [16] where an iterated sigma-point FastSLAM (ISP-FastSLAM) algorithm is proposed to minimize statistical linearization errors through the Gaussian–Newton iteration. A further application is presented by [17] where a walking robot uses sigma-point transformation for state estimation to guarantee stability in the system’s hybrid dynamics, which contains continuous and switching parts during movement. In [18], a vision-based SLAM system uses both extended Kalman filters (EKFs) and sigma-point Kalman filter (SPKF) algorithms and showed its superiority over the EKF. The pose estimation of mobile robots is discussed in [19] whereby several filter techniques like the Kalman filter (EKF), the unscented Kalman filter (UKF) and several variants of the particle filter (PF) are compared. It turns out that the UKF (also the sigma-point approach) exhibits almost the same computational cost. In [20], the inter-robot and robot–target correlations are discussed, and unscented transformation-based collaborative self-localization and a target tracking algorithm between robots are proposed. A tutorial on different approaches to exploit the structure of a system’s state and measurement models to reduce the computational demand of the algorithms is presented by [21]. In this publication, the computational complexity of different state estimation algorithms is presented, showing the superiority of the sigma-point transformation algorithms.

In all these publications, the problem of obstacle avoidance and/or the crossing problem in the presence of human actors are not taken into account, because of which the present paper is a further contribution to the robot–human interaction problem.

3. Computation of Intersections

The problem can be stated as follows:

A robot and human agent move in a common area according to their tasks or intentions. To avoid collisions, possible intersections of the paths of the agents should be predicted for both the trajectory planning and on-line interactions. To accomplish this, the positions, orientations and intended movements of the robot and human should be estimated as accurately as needed.

In this connection, uncertainties and noise on the random variables’ position/orientation x_R , x_H , ϕ_R and ϕ_H of the robot and human have a great impact on the calculation of the expected intersection position x_c . The random variable x_c is calculated as the crossing point of the extension of the orientation or velocity vectors of the robot and human, which may change during motion depending on the task and current interaction. The task is to calculate the intersection and its uncertainty in the presence of the known uncertainties of the acting agent robot and human.

System noise w_R and w_H for the robot and human can be obtained from experiments. The noise w_c of the “virtual” intersection is composed of the nonlinear transformed noise w_R and w_H and some additional noise v_c that may come from uncertainties of the nonlinear computation of the intersection position x_c (see Figure 1). In the following, the geometrical relations are described as well as the fuzzy approximations and nonlinear transformations of the random variables x_R , x_H , ϕ_R and ϕ_H .

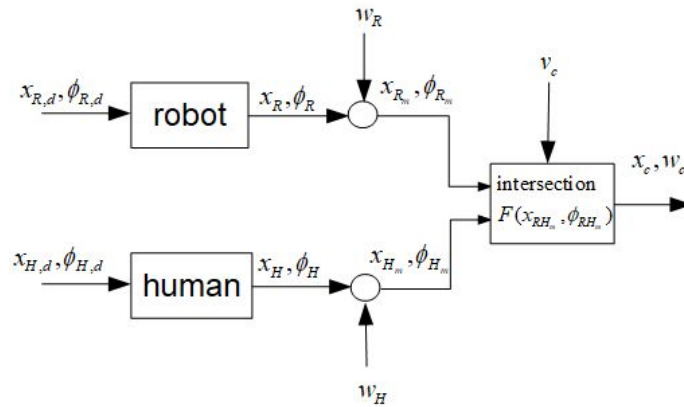


Figure 1. Intersection principle.

3.1. Geometrical Relations

Let the y -axis of the mobile coordinate frame of the robot and human be aligned with their directions of motion. Furthermore, let the orientation angles ϕ_R and ϕ_H of the robot and human be measured from the x -axis of the base frame counterclockwise. Let the intersection (x_c, y_c) of the two linear trajectories $\mathbf{x}_R(t)$ and $\mathbf{x}_H(t)$ in a plane be described by the following relations (see Figure 2):

$$\begin{aligned} x_H &= x_R + d_{RH} \cos(\phi_R + \delta_R) \\ y_H &= y_R + d_{RH} \sin(\phi_R + \delta_R) \\ x_R &= x_H + d_{RH} \cos(\phi_H + \delta_H) \\ y_R &= y_H + d_{RH} \sin(\phi_H + \delta_H) \end{aligned} \quad (1)$$

where $\mathbf{x}_H = (x_H, y_H)$ and $\mathbf{x}_R = (x_R, y_R)$ are the positions of the human and robot and ϕ_H and ϕ_R are their orientation angles, and δ_H and δ_R are the positive angles measured from the y coordinates counterclockwise. The angle at the intersection is $\tilde{\beta} = \pi - \delta_R - \delta_H$. The variables x_H , x_R , ϕ_R , ϕ_H , δ_H and δ_R ; distance d_{RH} ; and angle γ are assumed to be measurable. Angle γ is a bearing angle for the robot-to-human direction measured in base coordinates. If ϕ_H is not directly measurable, then it can be computed by

$$\phi_H = \arcsin((y_H - y_R)/d_{RH}) - \delta_H + \pi \quad (2)$$

The coordinates x_c and y_c of the intersection are computed straightforwardly by [8]

$$\begin{aligned} x_c &= \frac{A - B}{\tan \phi_R - \tan \phi_H} \\ y_c &= \frac{A \tan \phi_H - B \tan \phi_R}{\tan \phi_R - \tan \phi_H} \\ A &= x_R \tan \phi_R - y_R \\ B &= x_H \tan \phi_H - y_H \end{aligned} \quad (3)$$

Rewriting (3) leads to

$$\begin{aligned} x_c &= \left(x_R \frac{\tan \phi_R}{G} - y_R \frac{1}{G} \right) - \left(x_H \frac{\tan \phi_H}{G} - y_H \frac{1}{G} \right) \\ y_c &= \left(x_R \frac{\tan \phi_R \tan \phi_H}{G} - y_R \frac{\tan \phi_H}{G} \right) \\ &\quad - \left(x_H \frac{\tan \phi_H \tan \phi_R}{G} - y_H \frac{\tan \phi_R}{G} \right) \\ G &= \tan \phi_R - \tan \phi_H \end{aligned} \quad (4)$$

After rearranging (4), we observe that $\mathbf{x}_c = (x_c, y_c)^T$ is linear in $\mathbf{x}_{RH} = (x_R, y_R, x_H, y_H)^T$

$$\mathbf{x}_c = A_{RH} \cdot \mathbf{x}_{RH} \quad (5)$$

where

$$A_{RH} = f(\phi_R, \phi_H) = \frac{1}{G} \begin{pmatrix} \tan \phi_R & -1 & -\tan \phi_H & 1 \\ \tan \phi_R \tan \phi_H & -\tan \phi_H & -\tan \phi_R \tan \phi_H & \tan \phi_R \end{pmatrix}$$

This notation is of advantage for further computations, such as the fuzzification of the intersection problem and the transformation of the error distributions.

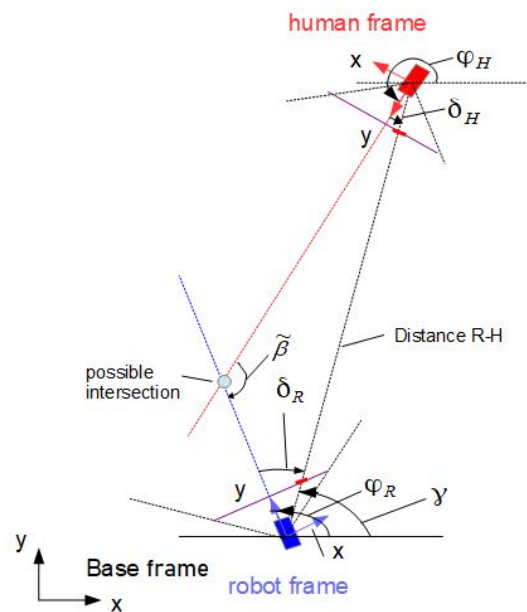


Figure 2. Human–robot scenario: geometry.

3.2. Computation of Intersections—Fuzzy Approach

The fuzzy solution presented in the following is a combination of classical analytical (crisp) methods and rule-based methods in the sense of a Takagi–Sugeno fuzzy rule base. An appropriate choice of the number of fuzzy sets and corresponding fuzzy rules depends strongly on the specific application. In the present case, fuzzy sets are used as the approximation of nonlinear functions. In the following, we introduce a fuzzy rule-based approximation of (5) with $n \times n$ fuzzy rules $R_{i,j}$

$$R_{i,j}: \text{ IF } \phi_R = \Phi_{Ri} \text{ AND } \phi_H = \Phi_{Hj} \quad (6) \\ \text{ THEN } \mathbf{x}_c = A_{RHij} \cdot \mathbf{x}_{RH}$$

n —the number of fuzzy terms, Φ_{Ri} and Φ_{Hj} for ϕ_R and ϕ_H , with the result

$$\mathbf{x}_c = \sum_{i,j} w_i(\phi_R) w_j(\phi_H) \cdot A_{RHij} \cdot \mathbf{x}_{RH} \quad (7)$$

$i, j = 1 \dots n, w_i(\phi_R), w_j(\phi_H) \in [0, 1]$ are normalized membership functions with $\sum_i w_i(\phi_R) = 1$ and $\sum_j w_j(\phi_H) = 1$.

Let the universes of discourse for ϕ_R and ϕ_H be $\phi_R, \phi_H \in [0, 360]$. Furthermore, let these universes of discourse be divided into n partitions (for example, 6) of 60, which leads to 6×6 fuzzy rules. The corresponding membership functions are shown in Figure 3. It turns out that this resolution leads to a poor fuzzy approximation. The approximation

quality can be improved by increasing the number of fuzzy sets, which however results in a quadratic increase in the number of fuzzy rules. To avoid an “explosion” of the number of fuzzy rules being computed in one time step, a set of sub-areas covering a small number of rules for each sub-area is defined. Based on the measurements of ϕ_R and ϕ_H , the appropriate sub-area is selected together with a corresponding set of rules (see Figure 4, sub-area A_R, A_H). With this, the number of rules to be activated at one time step of calculation is low, although the total number of rules can be high. At the borderlines between the sub-areas, abrupt changes may occur, which can be avoided by overlapping the sub-areas.

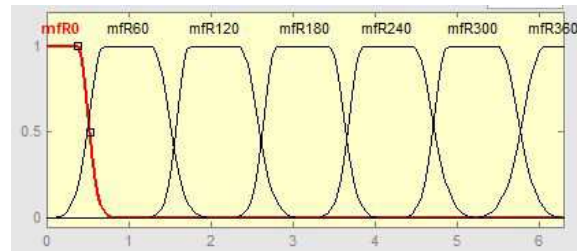


Figure 3. Membership functions for $\Delta\phi_R, \Delta\phi_H = 0 - 360^\circ$.

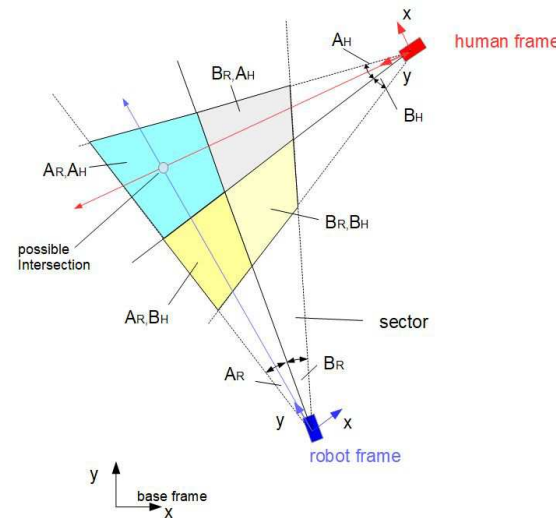


Figure 4. Fuzzy sectors.

3.3. Differential Approach

The positions and orientations of robots and humans are usually corrupted with noise originated from system uncertainties, sensor errors and motor characteristics. These uncertainties become apparent in uncertainties in the crossing/intersection areas of the trajectories of the robot and human. The analysis of uncertainty and noise at x_c generated by the noise at ϕ_R, ϕ_H and $x_{RH} = (x_R, y_R, x_H, y_H)^T$ requires a linearization of (4) around the operating points and with this a differential strategy. Let, for simplification, only the orientation angles ϕ_R and ϕ_H be corrupted with noise. In Section 4.3, the positions $x_{RH} = (x_R, y_R, x_H, y_H)^T$ are taken into account, too.

Differentiating (4) with $x_{RH} = const.$ yields

$$\begin{aligned} d\mathbf{x}_c &= \tilde{\mathbf{J}} \cdot d\mathbf{CE} \\ d\mathbf{CE} &= (d\phi_R \quad d\phi_H)^T; \quad \tilde{\mathbf{J}} = \begin{pmatrix} \tilde{J}_{11} & \tilde{J}_{12} \\ \tilde{J}_{21} & \tilde{J}_{22} \end{pmatrix} \end{aligned} \quad (8)$$

where

$$\begin{aligned}\tilde{J}_{11} &= \begin{pmatrix} -\tan \phi_H & 1 & \tan \phi_H & -1 \end{pmatrix} \frac{x_{RH}}{G^2 \cdot \cos^2 \phi_R} \\ \tilde{J}_{12} &= \begin{pmatrix} \tan \phi_R & -1 & -\tan \phi_R & 1 \end{pmatrix} \frac{x_{RH}}{G^2 \cdot \cos^2 \phi_H} \\ \tilde{J}_{21} &= \tilde{J}_{11} \cdot \tan \phi_H \\ \tilde{J}_{22} &= \tilde{J}_{12} \cdot \tan \phi_R\end{aligned}$$

The following sections deal with the accuracy of the computed intersection in the case of noisy orientation information (see Figure 5).

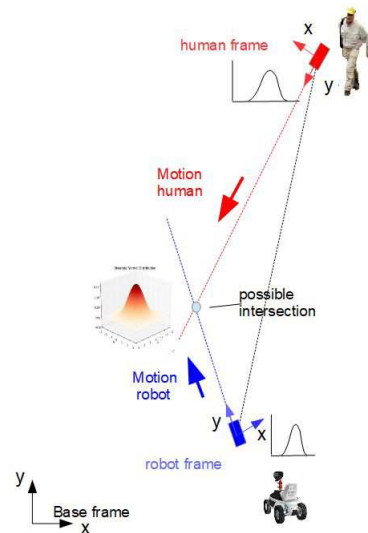


Figure 5. Intersection with noisy orientations.

4. Transformation of Gaussian Distributions

4.1. General Assumptions

Consider a nonlinear system

$$\mathbf{z} = F(\mathbf{x}) \quad (9)$$

where the random variables $\mathbf{x} = (x_1, x_2)^T$ denote the input, $\mathbf{z} = (z_1, z_2)^T$ denotes the output and F denotes a nonlinear transformation. The distribution of the uncorrelated Gaussian distributed components x_1 and x_2 is described by

$$f_{x_1, x_2} = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}} \exp\left(-\frac{1}{2}\left(\frac{e_{x_1}^2}{\sigma_{x_1}^2} + \frac{e_{x_2}^2}{\sigma_{x_2}^2}\right)\right) \quad (10)$$

where $e_{x_1} = x_1 - \bar{x}_1$, with \bar{x}_1 —the mean (x_1) and σ_{x_1} —the standard deviation x_1 , and $e_{x_2} = x_2 - \bar{x}_2$, with \bar{x}_2 —the mean (x_2) and σ_{x_2} —the standard deviation x_2 .

The goal is as follows: Given the nonlinear transformation (9) and the distribution (10), compute the output signals z_1 and z_2 and their distributions together with their standard deviations and the correlation coefficient. Linear systems transform Gaussian distributions linearly such that the output signals are also Gaussian-distributed. This does not apply for nonlinear systems, but if the input standard deviation is small enough, then a local linear transfer function can be built for which the outputs are Gaussian-distributed. Suppose the input standard deviations are small with respect to the nonlinear function, then the output distribution can be written as follows:

$$f_{z_1, z_2} = \frac{1}{2\pi\sigma_{z_1}\sigma_{z_2}\sqrt{1-\rho_{z_{12}}^2}} \cdot \exp\left(-\frac{1}{2(1-\rho_{z_{12}}^2)}\left(\frac{e_{z_1}^2}{\sigma_{z_1}^2} + \frac{e_{z_2}^2}{\sigma_{z_2}^2} - \frac{2\rho_{z_{12}}e_{z_1}e_{z_2}}{\sigma_{z_1}\sigma_{z_2}}\right)\right) \tag{11}$$

$\rho_{z_{12}}$ —the correlation coefficient.

4.2. Statistical Linearization, Two Inputs–Two Outputs

Let the nonlinear transformation F be described by two smooth transfer functions (see block scheme Figure 6)

$$\begin{aligned} z_1 &= f_1(x_1, x_2) \\ z_2 &= f_2(x_1, x_2) \end{aligned} \tag{12}$$

where $(x_1, x_2) = (\phi_R, \phi_H)$ and $(z_1, z_2) = (x_c, y_c)$.

The linearization of (12) yields

$$dz = \tilde{J} \cdot dx \quad \text{or} \quad e_z = \tilde{J} \cdot e_x \tag{13}$$

with

$$\begin{aligned} e_z &= (e_{z_1}, e_{z_2})^T \quad \text{and} \quad e_x = (e_{x_1}, e_{x_2})^T \\ dz &= (dz_1, dz_2)^T \quad \text{and} \quad dx = (dx_1, dx_2)^T \end{aligned} \tag{14}$$

$$\tilde{J} = \begin{pmatrix} \partial f_1 / \partial x_1 & \partial f_1 / \partial x_2 \\ \partial f_2 / \partial x_1 & \partial f_2 / \partial x_2 \end{pmatrix} \tag{15}$$

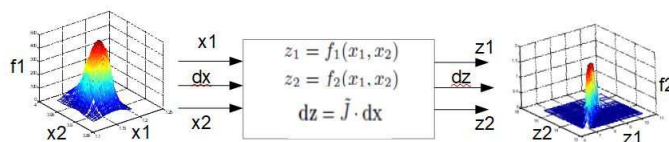


Figure 6. Differential transformation.

4.2.1. Output Distribution

To obtain the density f_{z_1, z_2} (11) of the output signal, we invert (15) and substitute the entries of e_x into (10). \tilde{J} is invertible if it is positive definite with $|\tilde{J}| > 0$. Otherwise, there exist singularities due to different constellations of the position vector x_{RH} and/or the orientations ϕ_R and ϕ_H . To find all the singularities requires a further analysis, which is not the content of this paper. However, a simple heuristic leads us to some obvious situations: If $\phi_R = \phi_H$ or $\phi_R = \phi_H + \pi$, then the human and robot would move in parallel either in the same or the opposite direction. On the other hand, one may also obtain diverging trajectories with no crossing.

$$e_x = J \cdot e_z \tag{16}$$

with $J = \tilde{J}^{-1}$ and

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} = \begin{pmatrix} j_{xz} \\ j_{yz} \end{pmatrix} \tag{17}$$

where $j_{xz} = (J_{11}, J_{12})$ and $j_{yz} = (J_{21}, J_{22})$. The entries J_{ij} are the result of the inversion of \tilde{J} . From this substitution, we obtain

$$f_{x_1, x_2} = K_{x_1, x_2} \cdot \exp\left(-\frac{1}{2} \cdot \mathbf{e}_z^T \cdot (\mathbf{j}_{x_1, z}^T, \mathbf{j}_{x_2, z}^T) \cdot S_x^{-1} \cdot \begin{pmatrix} \mathbf{j}_{x_1, z} \\ \mathbf{j}_{x_2, z} \end{pmatrix} \cdot \mathbf{e}_z\right) \quad (18)$$

where $K_{x_1, x_2} = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}}$ and

$$S_x^{-1} = \begin{pmatrix} \frac{1}{\sigma_{x_1}^2}, 0 \\ 0, \frac{1}{\sigma_{x_2}^2} \end{pmatrix} \quad (19)$$

The exponent of (18) is rewritten into

$$x_{po} = -\frac{1}{2} \cdot \left(\frac{1}{\sigma_{x_1}^2} (e_{z_1} J_{11} + e_{z_2} J_{12})^2 + \frac{1}{\sigma_{x_2}^2} (e_{z_1} J_{21} + e_{z_2} J_{22})^2\right) \quad (20)$$

and furthermore

$$x_{po} = -\frac{1}{2} \cdot \left[e_{z_1}^2 \left(\frac{J_{11}^2}{\sigma_{x_1}^2} + \frac{J_{21}^2}{\sigma_{x_2}^2}\right) + e_{z_2}^2 \left(\frac{J_{12}^2}{\sigma_{x_1}^2} + \frac{J_{22}^2}{\sigma_{x_2}^2}\right) + 2 \cdot e_{z_1} e_{z_2} \left(\frac{J_{11} J_{12}}{\sigma_{x_1}^2} + \frac{J_{21} J_{22}}{\sigma_{x_2}^2}\right)\right] \quad (21)$$

Let

$$A = \left(\frac{J_{11}^2}{\sigma_{x_1}^2} + \frac{J_{21}^2}{\sigma_{x_2}^2}\right); \quad B = \left(\frac{J_{12}^2}{\sigma_{x_1}^2} + \frac{J_{22}^2}{\sigma_{x_2}^2}\right) \\ C = \left(\frac{J_{11} J_{12}}{\sigma_{x_1}^2} + \frac{J_{21} J_{22}}{\sigma_{x_2}^2}\right) \quad (22)$$

then a comparison of x_{po} in (21) and the exponent in (11) yields

$$\frac{1}{(1 - \rho_{z_{12}}^2) \sigma_{z_1}^2} = A; \quad \frac{1}{(1 - \rho_{z_{12}}^2) \sigma_{z_2}^2} = B \\ \frac{-2\rho_{z_{12}}}{(1 - \rho_{z_{12}}^2) \sigma_{z_1} \sigma_{z_2}} = 2C \quad (23)$$

The standard deviations σ_{z_1} and σ_{z_2} and the correlation coefficient $\rho_{z_{12}}$ yield

$$\rho_{z_{12}} = -\frac{C}{\sqrt{AB}} \\ \frac{1}{\sigma_{z_1}^2} = A - \frac{C^2}{B}; \quad \frac{1}{\sigma_{z_2}^2} = B - \frac{C^2}{A} \quad (24)$$

The result is as follows: If the parameter of the input distribution and the transfer function $F(x, y)$ are known, then the output distribution parameters can be computed straightforwardly.

4.2.2. Fuzzy Solution

To save computing costs in real time, we create a TS fuzzy model that is represented by the rules R_{ij} .

$$\begin{aligned}
 &R_{ij} : \\
 \text{IF } &x_1 = X_{1i} \text{ AND } x_2 = X_{2i} \\
 \text{THEN } &\rho_{z_{12}} = -\frac{C_{ij}}{\sqrt{A_{ij}B_{ij}}} \\
 \text{AND} & \\
 &\frac{1}{\sigma_{z_1}^2} = A_{ij} - \frac{C_{ij}^2}{B_{ij}}; \\
 \text{AND} & \\
 &\frac{1}{\sigma_{z_2}^2} = B_{ij} - \frac{C_{ij}^2}{A_{ij}}
 \end{aligned} \tag{25}$$

where X_{1i}, X_{2i} are fuzzy terms for x_1, x_2 , and A_{ij}, B_{ij}, C_{ij} are functions of the predefined variables $x_1 = x_{1i}$ and $x_2 = x_{2i}$.

From (25), we derive

$$\begin{aligned}
 \rho_{z_{12}} &= -\sum_{ij} w_i(x_1)w_j(x_2) \frac{C_{ij}}{\sqrt{A_{ij}B_{ij}}} \\
 \frac{1}{\sigma_{z_1}^2} &= \sum_{ij} w_i(x_1)w_j(x_2) \left(A_{ij} - \frac{C_{ij}^2}{B_{ij}} \right) \\
 \frac{1}{\sigma_{z_2}^2} &= \sum_{ij} w_i(x_1)w_j(x_2) \left(B_{ij} - \frac{C_{ij}^2}{A_{ij}} \right)
 \end{aligned} \tag{26}$$

$w_i(x_1) \in [0, 1]$ and $w_j(x_2) \in [0, 1]$ are the weighting functions with $\sum_i w_i(x_1) = 1$, $\sum_j w_j(x_2) = 1$.

4.2.3. Inverse Solution

The previous paragraph discussed the direct transformation task: Let the distribution parameters of the input variable be defined and find the corresponding output parameters. However, it might also be useful to solve the inverse task: Given the output parameters (standard deviation and correlation coefficient), find the corresponding input parameters. This solution of the inverse task is similar to those discussed in Section 4.2. The starting points are equations (10) and (11), which describe the distributions of the inputs and outputs, respectively. Then, we substitute (13) into (10) and rename the resulting exponent $x_{p_{o_z}}$ into $x_{p_{o_x}}$ and discuss the exponent $x_{p_{o_x}}$

$$x_{p_{o_x}} = \frac{-1}{2(1 - \rho_{z_{12}}^2)} \left(\mathbf{e}_x^T \tilde{\mathbf{f}}^T S_z^{-1} \tilde{\mathbf{f}} \mathbf{e}_x - \frac{2\rho_{z_{12}}e_{z_1}e_{z_2}}{\sigma_{z_1}\sigma_{z_2}} \right) \tag{27}$$

with

$$S_x^{-1} = \begin{pmatrix} \frac{1}{\sigma_{z_1}^2}, 0 \\ 0, \frac{1}{\sigma_{z_2}^2} \end{pmatrix}$$

Now, comparing (27) with the exponent of (10) of the input density, we find that the mixed term in (27) must be zero, from which we obtain the correlation coefficient $\rho_{z_{12}}$ and with this the standard deviations of the inputs

$$\begin{aligned} \rho_{z_{12}} &= \left(\frac{\tilde{J}_{11}\tilde{J}_{12}}{\sigma_{z_1}^2} + \frac{\tilde{J}_{21}\tilde{J}_{22}}{\sigma_{z_2}^2} \right) \frac{\sigma_{z_1}\sigma_{z_2}}{(\tilde{J}_{11}\tilde{J}_{22} + \tilde{J}_{12}\tilde{J}_{21})} \\ \frac{1}{\sigma_x^2} &= \left(\frac{\tilde{J}_{11}^2}{\sigma_{z_1}^2} + \frac{\tilde{J}_{21}^2}{\sigma_{z_2}^2} - \frac{2\rho_{z_{12}}}{\sigma_{z_1}\sigma_{z_2}} \tilde{J}_{11}\tilde{J}_{21} \right) / (1 - \rho_{z_{12}}^2) \\ \frac{1}{\sigma_y^2} &= \left(\frac{\tilde{J}_{12}^2}{\sigma_{z_1}^2} + \frac{\tilde{J}_{22}^2}{\sigma_{z_2}^2} - \frac{2\rho_{z_{12}}}{\sigma_{z_1}\sigma_{z_2}} \tilde{J}_{12}\tilde{J}_{22} \right) / (1 - \rho_{z_{12}}^2) \end{aligned} \tag{28}$$

The detailed development can be found in [22].

4.3. Six Inputs–Two Outputs

Consider again the nonlinear system

$$\mathbf{x}_c = F(\mathbf{x}) \tag{29}$$

In the previous subsections, we assumed the positions \mathbf{x}_R and \mathbf{x}_H not to be corrupted with noise. However, taking into account the positions to be random variables, the number of inputs is 6 so that the input vector yields $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)^T$ or $\mathbf{x} = (\phi_R, \phi_H, x_R, y_R, x_H, y_H)$ with the output vector $\mathbf{x}_c = (x_c, y_c)^T$.

Furthermore, let the uncorrelated Gaussian-distributed inputs $x_1 \dots x_6$ be described by the 6-dim density

$$f_{x_i} = \frac{1}{(2\pi)^{6/2} |S_x|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{e}_x^T S_x^{-1} \mathbf{e}_x)\right) \tag{30}$$

where $\mathbf{e}_x = (e_{x_1}, e_{x_2}, \dots, e_{x_6})^T$; $\mathbf{e}_x = \mathbf{x} - \bar{\mathbf{x}}$, $\bar{\mathbf{x}}$ —the mean(\mathbf{x}) and S_x —the covariance matrix.

$$S_x = \begin{pmatrix} \sigma_{x_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{x_2}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \sigma_{x_6}^2 \end{pmatrix}$$

According to (11), the output density is described by

$$\begin{aligned} f_{x_c, y_c} &= \frac{1}{2\pi\sigma_{x_c}\sigma_{y_c}\sqrt{1-\rho^2}} \cdot \\ \exp\left(-\frac{1}{2(1-\rho^2)}(\mathbf{e}_{x_c}^T S_c^{-1} \mathbf{e}_{x_c} - \frac{2\rho e_{x_c} e_{y_c}}{\sigma_{x_c}\sigma_{y_c}})\right) \end{aligned} \tag{31}$$

ρ —the correlation coefficient, $\mathbf{e}_{x_c} = (e_{x_c}, e_{y_c})^T$.

After some calculations [23], we find for ρ , $\frac{1}{\sigma_{x_c}^2}$ and $\frac{1}{\sigma_{y_c}^2}$

$$\begin{aligned} \rho &= -\frac{C}{\sqrt{AD}} \\ \frac{1}{\sigma_{x_c}^2} &= A - \frac{C^2}{D}; \quad \frac{1}{\sigma_{y_c}^2} = D - \frac{C^2}{A} \end{aligned} \tag{32}$$

with

$$\begin{aligned} A &= \sum_{i=1}^6 \frac{1}{\sigma_{x_i}^2} J_{i1}^2; \quad B = \sum_{i=1}^6 \frac{1}{\sigma_{x_i}^2} J_{i1} J_{i2} \\ C &= \sum_{i=1}^6 \frac{1}{\sigma_{x_i}^2} J_{i1} J_{i2}; \quad D = \sum_{i=1}^6 \frac{1}{\sigma_{x_i}^2} J_{i2}^2 \end{aligned} \tag{33}$$

This is the counterpart to the 2-dim input case (24).

4.3.1. Inverse Solution

An inverse solution cannot be uniquely computed due to the undetermined character of the 6-input–2-output system. Therefore, from the required variances at the intersection position (output), the corresponding variances for the positions and orientations of the robot–human or robot–robot (input) cannot be concluded.

4.3.2. Fuzzy Approach

The steps to the fuzzy approach are very similar to those of the 2-input case:

- Define the operation points $\mathbf{x}_i = (x_1, x_2, x_3, x_4, x_5, x_6)_i^T$;
- Compute A_i , B_i and C_i at $\mathbf{x}_i = (x_1, x_2, x_3, x_4, x_5, x_6)_i^T$ from (33);
- Formulate the fuzzy rules R_i according to (25) and (26), $i = 1 \dots n$.

The number n of rules is computed as follows:

With $l = 6$ —the number of fuzzy terms and $k = 6$ —the number of inputs, we obtain $n = l^k = 6^6$ —the number of rules.

This number of rules is unacceptably high. To limit n to an adequate number, one has to limit the number of inputs and/or fuzzy terms to look for the most influential variables either in a heuristic or systematic way [24]. This however is not the issue to be discussed in this paper.

5. Sigma-Point Transformation

In the following, the estimation/identification of the standard deviations of possible intersection coordinates of trajectories for both the robot–robot and human–robot combinations by means of the sigma-point technique is discussed. The following method is based on the unscented Kalman filter technique where the intersections cannot be directly measured but predicted/computed only. Nevertheless, it is possible to compute the variance of the predicted events, such as possible collisions or planned rendezvous situations, by a direct propagation of statistical parameters—the sigma points—through the nonlinear geometrical relation, which is a result of the crossing of two trajectories. Let $\mathbf{x} = (x_1, x_2)^T$ —the input vector and $\mathbf{x}_c = (x_{c1}, x_{c2})^T$ —the output vector where for the special case $(x_1, x_2)^T = (\phi_R, \phi_H)^T$ and $(x_{c1}, x_{c2})^T = (x_c, y_c)^T$. The nonlinear relation between \mathbf{x} and \mathbf{x}_c is given by (34)

$$\mathbf{x}_c = \mathbf{F}(\mathbf{x}) \quad (34)$$

For the discrete case, we obtain for the state \mathbf{x}_c

$$\mathbf{x}_c(k) = \mathbf{F}(\mathbf{x}(k-1) + \mathbf{w}(k-1)) \quad (35)$$

and for the measured output $\mathbf{z}_c(k)$

$$\mathbf{z}_c(k) = \mathbf{h}(\mathbf{x}_c)(k) + \mathbf{v}(k) \quad (36)$$

where \mathbf{w} and \mathbf{v} are the system noise and measurement noise, respectively. $\mathbf{h}(\mathbf{x}_c)$ is the output nonlinearity. Furthermore, let there be the following:

$\bar{\mathbf{x}}(k)$ —the mean at time t_k ;

$\mathbf{P}(k)$ —the covariance matrix;

\mathbf{x}_0 —the initial state with the known mean $\mu_0 = E(\mathbf{x}_0)$;

$\mathbf{P}_0(k) = E[(\mathbf{x}_0 - \mu_0)(\mathbf{x}_0 - \mu_0)^T]$.

5.1. Selection of Sigma Points

Sigma points are the selected parameters of a given error distribution of a random variable. Sigma points lie along the major eigen-axes of the covariance matrix of the random variable. The height of each sigma point (see Figure 7) represents its relative weight W^j used in the following selection procedure.

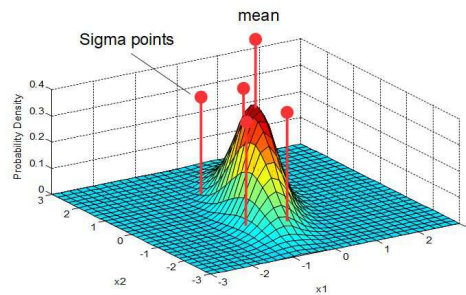


Figure 7. Sigma points for a 2-dim Gaussian random variable.

Let $\mathbf{X}(k-1)$ be a set of $2n+1$ sigma points where n is the dimension of the state space (in our example, $n=2$).

$$\mathbf{X}(k-1) = \{(\mathbf{x}^j(k-1), W^j) | j = 0 \dots 2n\} \quad (37)$$

Consider the following selection of sigma points

$$\begin{aligned} \mathbf{x}^0(k-1) &= \bar{\mathbf{x}}(k-1) \\ -1 < W^0 < 1 \\ W^0 &= \frac{\lambda}{n+\lambda}; \quad \lambda = \alpha^2(n+\kappa) - n \\ \mathbf{x}^i(k-1) &= \bar{\mathbf{x}}(k-1) + \sqrt{\left(\frac{n}{1-W^0}\mathbf{P}(k-1)\right)}; \quad i = 1 \dots n \\ \mathbf{x}^i(k-1) &= \bar{\mathbf{x}}(k-1) - \sqrt{\left(\frac{n}{1-W^0}\mathbf{P}(k-1)\right)}; \quad i = (n+1) \dots 2n \\ W^j &= \frac{1-W^0}{2n} \end{aligned} \quad (38)$$

under the following condition

$$\sum_{j=0}^{2n} W^j = 1 \quad (40)$$

α and κ are scaling factors. A usual choice is $\alpha = 10^{-2}$ and $\kappa = 0$. $\sqrt{\frac{n}{1-W^0}\mathbf{P}(k-1)}$ is the row/column of the matrix square root of $\frac{n}{1-W^0}\mathbf{P}$. The square root of a matrix \mathbf{P} is the solution \mathbf{S} for $\mathbf{P} = \mathbf{S} \cdot \mathbf{S}$, which is obtained by Cholesky factorization.

5.2. Model Forecast Step

To go on with the UKF, the following step is devoted to the model forecast. In this way, the sigma points $\mathbf{x}^j(k)$ are propagated through the nonlinear process model

$$\mathbf{x}_c^{f,j}(k) = \mathbf{F}(\mathbf{x}^j(k-1)) \quad (41)$$

where the superscript f means “forecast”. From these transformed and forecasted sigma points, the mean and covariance for the forecast value of $\mathbf{x}_c(k)$ are

$$\begin{aligned} \mathbf{x}_c^f(k) &= \sum_{j=0}^{2n} W^j \mathbf{x}_c^{f,j}(k) \\ \mathbf{P}^f(k) &= \sum_{j=0}^{2n} W^j (\mathbf{x}_c^{f,j}(k) - \mathbf{x}_c^f(k)) (\mathbf{x}_c^{f,j}(k) - \mathbf{x}_c^f(k))^T \end{aligned} \quad (42)$$

5.3. Measurement Update Step

In this step, the sigma points are propagated through the nonlinear observation model

$$\mathbf{z}_c^{f,j}(k) = \mathbf{h}(\mathbf{x}_c^j(k-1)) \quad (43)$$

from which we obtain the mean and covariance (innovation covariance)

$$\begin{aligned} \mathbf{z}_c^f(k-1) &= \sum_{j=0}^{2n} W^j \mathbf{z}_c^{f,j}(k-1) \\ \text{Cov}(\tilde{\mathbf{z}}_c^f(k-1)) &= \\ &\sum_{j=0}^{2n} W^j (\mathbf{z}_c^{f,j}(k-1) - \mathbf{z}_c^f(k-1)) \times \\ &(\mathbf{z}_c^{f,j}(k-1) - \mathbf{z}_c^f(k-1))^T + \mathbf{R}(k) \end{aligned} \quad (44)$$

and the cross-covariance

$$\begin{aligned} \text{Cov}(\tilde{\mathbf{x}}_c^f(k), \tilde{\mathbf{z}}_c^f(k-1)) &= \\ &\sum_{j=0}^{2n} W^j (\mathbf{x}_c^{f,j}(k) - \mathbf{x}_c^f(k)) (\mathbf{z}_c^{f,j}(k-1) - \mathbf{z}_c^f(k-1))^T \end{aligned} \quad (45)$$

5.4. Data Assimilation Step

In this step, the forecast information is combined with the new information from the output $\mathbf{z}(k)$ from which we obtain, with the Kalman filter, gain \mathbf{K}

$$\hat{\mathbf{x}}_c(k) = \mathbf{x}_c^f(k) + \mathbf{K}(k)(\mathbf{z}_c(k) - \mathbf{z}_c^f(k-1)) \quad (46)$$

The gain \mathbf{K} is given by

$$\mathbf{K}(k) = \text{Cov}(\tilde{\mathbf{x}}_c^f(k), \tilde{\mathbf{z}}_c^f(k-1)) \cdot \text{Cov}^{-1}(\tilde{\mathbf{z}}_c^f(k-1)) \quad (47)$$

and the posterior covariance is updated by

$$\mathbf{P}(k) = \mathbf{P}^f(k) - \mathbf{K}(k) \cdot \text{Cov}(\tilde{\mathbf{z}}_c^f(k-1)) \mathbf{K}^T(k) \quad (48)$$

Usually, it is sufficient to compute the mean and variance for the output/state \mathbf{x}_c of the nonlinear static system $\mathbf{F}(\mathbf{x})$. In this case, it is possible to stop further computing at Equation (42), meaning to rather calculate the transformed sigma points $\mathbf{x}_c^{f,j}$ and develop the specific output means and variances from (41) and (42). In this connection, it is enough to substitute the covariance matrix \mathbf{Q} into (38) instead of \mathbf{P} . One advantage of the sigma-point approach prior to statistical linearization is the easy scalability to multi-dimensional random variables.

For the intersection problem, there are 2 cases:

1. The 2 inputs, 2 outputs (2 orientation angles and 2 crossing coordinates);
2. The 6 inputs, 2 outputs (2 orientation angles and 4 position coordinates, and 2 crossing coordinates).

For the statistical linearization (method 1), the step from the 2 inputs–2 outputs case to the (6,2)-case is computationally more costly than that for the sigma-point approach (method 2), (see Equations (20)–(24) versus Equations (37) and (40)–(42)).

5.5. Sigma Points—Fuzzy Solutions

In order to lower the computing effort, the application of the TS fuzzy interpolation may be a solution, which will be shown in the following. Having a look at the two-

dimensional problem, we can see a nonlinear propagation of the input sigma points through a nonlinear function \mathbf{F} . Let \mathbf{x}^j be the two-dimensional “input” sigma points

$$\mathbf{x}^j = (x_1^j, x_2^j)^T \quad (49)$$

or for the special case “intersection”

$$\mathbf{x}^j = (\phi_R^j, \phi_H^j)^T \quad (50)$$

The propagation through \mathbf{F} leads to the “output” sigma points

$$\mathbf{x}_c^{f,j}(k) = \mathbf{F}(\mathbf{x}^j(k-1)) \quad (51)$$

or for the special case

$$\begin{aligned} \mathbf{x}_c^{f,j}(k) &= \mathbf{F}(x_1^j(k-1), x_2^j(k-1)) = \\ &\mathbf{F}(\phi_R^j(k-1), \phi_H^j(k-1)) \end{aligned} \quad (52)$$

The special nonlinear function \mathbf{F} is described by (see (5))

$$\mathbf{x}_c = \mathbf{A}_{RH}(\phi_R, \phi_H) \cdot \mathbf{x}_{RH} \quad (53)$$

where \mathbf{A}_{RH} is a nonlinear matrix (6) linearly combined with the position vector $\mathbf{x}_{RH} = (x_R, y_R, x_H, y_H)^T$.

A fuzzification aims at \mathbf{A}_{RH} :

$$\begin{aligned} \mathbf{F}^{fuzz}(\phi_R, \phi_H) &= \mathbf{A}_{RH}^{fuzz} \cdot \mathbf{x}_{RH} = \\ &\sum_{l_1, l_2}^m w^{l_1}(\phi_R) w^{l_2}(\phi_H) \cdot \mathbf{A}_{RH}(\phi_R^{l_1}, \phi_H^{l_2}) \cdot \mathbf{x}_{RH} \end{aligned} \quad (54)$$

Applied to the sigma points (ϕ_R^j, ϕ_H^j) , we obtain a TS fuzzy model described by the following rules R_{l_1, l_2}

$$\begin{aligned} &R_{l_1, l_2} : \\ &IF \quad \phi_R^j = \Phi_{Rl_1}^j \quad AND \quad \phi_H^j = \Phi_{Hl_2}^j \\ &THEN \quad \mathbf{x}_c^{f,j} = \mathbf{A}_{RH}(\phi_R^{l_1, j}, \phi_H^{l_2, j}) \cdot \mathbf{x}_{RH} \end{aligned} \quad (55)$$

where $\Phi_{Rl_1}^j, \Phi_{Hl_2}^j$ are fuzzy terms for ϕ_R^j, ϕ_H^j ; the matrices \mathbf{A}_{RH} are functions of the predefined variables ϕ_R^j and ϕ_H^j . This set of rules leads to the result

$$\begin{aligned} \mathbf{x}_c^{f,j} &= \mathbf{F}^{fuzz}(\phi_R^j, \phi_H^j) = \\ &\sum_{l_1, l_2}^m w^{l_1}(\phi_R^j) w^{l_2}(\phi_H^j) \cdot \mathbf{A}_{RH}(\phi_R^{l_1, j}, \phi_H^{l_2, j}) \cdot \mathbf{x}_{RH} \end{aligned} \quad (56)$$

$w^1(\phi_R^j) \in [0, 1]$ and $w^2(\phi_H^j) \in [0, 1]$ are weighting functions with $\sum_{l_1} w^1 = 1$, $\sum_{l_2} w^2 = 1$. The advantage of this approach is that the $l_1 \times l_2$ matrices $\mathbf{A}_{RH}^{l_1, l_2, j} = \mathbf{A}_{RH}(\phi_R^{l_1, j}, \phi_H^{l_2, j})$ can be computed off-line. Then, the calculation of the mean and covariance matrix is obtained by

$$\begin{aligned} \mathbf{x}_c^f(k) &= \sum_{j=0}^{2n} \mathbf{W}^j \mathbf{x}_c^{f,j}(k) \\ \mathbf{P}^f(k) &= \sum_{j=0}^{2n} \mathbf{W}^j \tilde{\mathbf{x}}_c^{f,j}(k) (\tilde{\mathbf{x}}_c^{f,j}(k))^T \\ \tilde{\mathbf{x}}_c^{f,j} &= \mathbf{x}_c^{f,j} - \mathbf{x}_c^f \end{aligned} \quad (57)$$

From the covariance \mathbf{P}^f , the variances $\sigma_{c_{xx}}, \sigma_{c_{yy}}, \sigma_{c_{xy}}$ can be obtained

$$\begin{aligned} \sigma_{c_{xx}} &= E((x_c^f - \tilde{x}_c^f)^2) \\ \sigma_{c_{yy}} &= E((y_c^f - \tilde{y}_c^f)^2) \\ \sigma_{c_{xy}} = \sigma_{c_{yx}} &= E((x_c^f - \tilde{x}_c^f) \cdot (y_c^f - \tilde{y}_c^f)) \end{aligned} \quad (58)$$

5.6. Inverse Solution

The inverse solution for the sigma-point approach is much easier to obtain than that for the statistical linearization method. Starting from Equation (34), we build the inverse function

$$\mathbf{x} = \mathbf{F}^{-1}(\mathbf{x}_c) \quad (59)$$

on the condition that \mathbf{F}^{-1} exists. Then, the covariance matrix \mathbf{P} is defined in correspondence to the required variances $\sigma_{c_{xx}}, \sigma_{c_{yy}}$ and $\sigma_{c_{xy}}$. The following steps correspond to Equations (34)–(42). The position vector \mathbf{x}_{RH} is assumed to be known. The inversion of \mathbf{F} requires a linearization of \mathbf{x}_{RH} and a starting point to obtain a stable convergence to the inverse \mathbf{F}^{-1} . The result is the mean \mathbf{x} and the covariance \mathbf{Q} at the input. A reliable inversion is only possible for the 2-input–2-output case.

5.7. Six-Inputs–Two-Outputs

This case works exactly as the 2-input–2-output case along with Equations (34)–(42) due to the fact that the computation of the sigma points (38)–(40) and the propagation through the nonlinearity \mathbf{F} automatically include the input and output dimensions.

6. Simulation Results

The following simulations show the results of the uncertainties of the predicted intersections based on statistical linearization and sigma-point transformation. For both methods, identical parameters are employed for comparison reasons (see Figure 2). The position/orientation of the robot and human are given by the following:

$$\mathbf{x}_R = (x_R, y_R)^T = (2, 0)^T \text{ m};$$

$$\mathbf{x}_H = (x_H, y_H)^T = (4, 10)^T \text{ m};$$

$$\phi_R = 1.78 \text{ rad} = 102^\circ;$$

$$\phi_H = 3.69 \text{ rad} = 212^\circ.$$

ϕ_R and ϕ_H are corrupted by Gaussian noise with standard deviations (std) of $\sigma_{\phi_R} = \sigma_{x_1} = 0.02 \text{ rad}$, ($= 1.1^\circ$), and $\sigma_{\phi_H} = \sigma_{x_2} = 0.02 \text{ rad}$, ($= 1.1^\circ$).

6.1. Statistical Linearization

Table 1 shows a comparison of the non-fuzzy method with the fuzzy approach using sectors of $60^\circ, 30^\circ, 15^\circ, 7.5^\circ$ of the unit circle for the orientations of the robot and human. The notations in Table 2 are as follows: σ_{x_c} —std-computed, σ_{x_m} —std-measured, etc. As expected, we see that higher resolutions lead to a better match between the fuzzy and

analytical approach. Furthermore, the match between the measured and calculated values depends on the form of membership functions (MFS). For example, low input standard deviations (0.02 rad) show a better match for Gaussian membership functions, and higher input standard deviations (0.05 rad = 2.9°) require Gaussian bell-shaped membership functions, which comes from different smoothing effects (see columns 4 and 5 in Table 2).

A comparison of the control surfaces and corresponding measurements x_{cm}, y_{cm} (black and red dots) is depicted in Figures 8–10. Figure 8 shows the control surface of x_c and y_c for the non-fuzzy case (4). The control surfaces of the fuzzy approximations (7) for the 30° and 7.5° sectors are shown in Figures 9 and 10. The resolution 30° (Figure 9) shows a very high deviation compared to the non-fuzzy approach (Figure 8), which decreases further down to the resolution 7.5° (Figure 10). This explains the high differences between the measured and computed standard deviations and correlation coefficients, in particular for sector sizes of 30° and higher.

Table 1. Standard deviations and fuzzy and non-fuzzy results.

Input Std	0.02 Gauss, Bell Shaped (GB)				Gauss	0.05 GB
	60°	30°	15°	7.5°	7.5°	7.5°
sector size / °						
non-fuzz σ_{x_c}	0.143	0.140	0.138	0.125	0.144	0.366
fuzz σ_{x_c}	0.220	0.184	0.140	0.126	0.144	0.367
non-fuzz σ_{x_m}	0.160	0.144	0.138	0.126	0.142	0.368
fuzz σ_{x_m}	0.555	0.224	0.061	0.225	0.164	0.381
non-fuzz σ_{y_c}	0.128	0.132	0.123	0.114	0.124	0.303
fuzz σ_{y_c}	0.092	0.087	0.120	0.112	0.122	0.299
non-fuzz σ_{y_m}	0.134	0.120	0.123	0.113	0.129	0.310
fuzz σ_{y_m}	0.599	0.171	0.034	0.154	0.139	0.325
non-fuzz ρ_{xy_c}	0.576	0.541	0.588	0.561	0.623	0.669
fuzz ρ_{xy_c}	−0.263	0.272	0.478	0.506	0.592	0.592
non-fuzz ρ_{xy_m}	0.572	0.459	0.586	0.549	0.660	0.667
fuzz ρ_{xy_m}	0.380	0.575	0.990	0.711	0.635	0.592

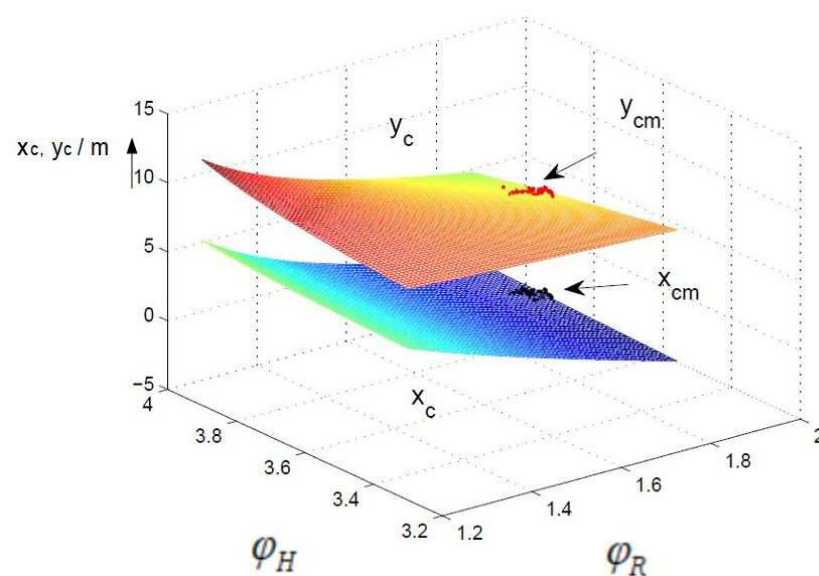


Figure 8. Control surface non-fuzzy, units of ϕ_R and ϕ_H in rad.

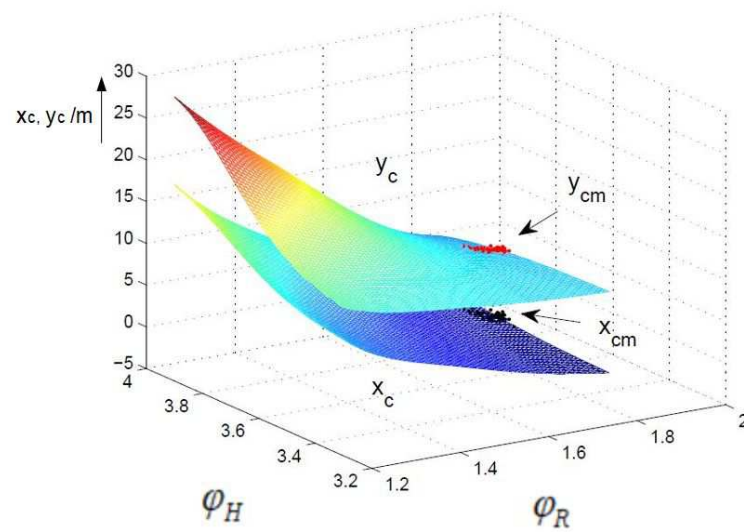


Figure 9. Control surface fuzzy, 30° , units of ϕ_R and ϕ_H in rad.

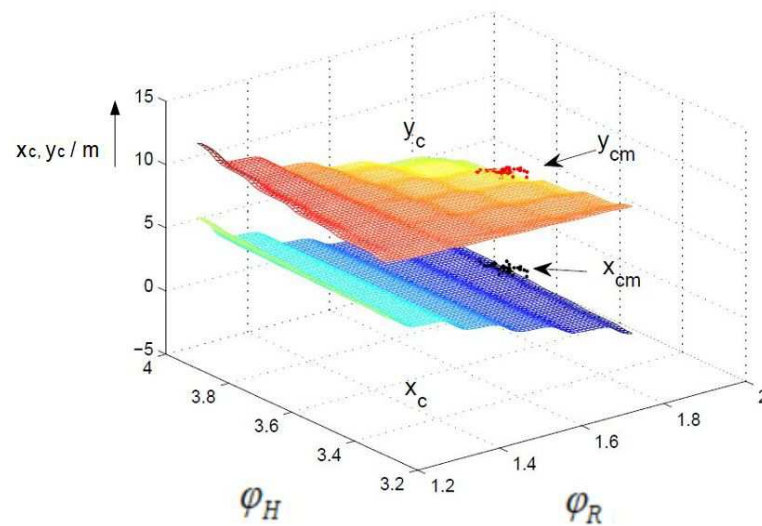


Figure 10. Control surface fuzzy, 7.5° , units of ϕ_R and ϕ_H in rad.

6.2. Sigma-Point Method

Two-inputs–two-outputs:

The simulation of the sigma-point method is based on a Matlab implementation of an unscented Kalman filter by [25]. The first example deals with the 2-inputs–2-outputs case in which only the orientations are taken into account, but the disturbances of the positions of the robot and human are not part of the sigma-point calculation. A comparison between the computed and measured covariance shows a very good match. The same holds for the standard deviations $\sigma_{x_c}, \sigma_{y_c}$. A comparison with the statistical linearization shows a good match as well (see Table 2, rows 1 and 2).

A view at the sigma points presents the following results: Figure 11 shows the two-dimensional distribution of the orientation angles (ϕ_R, ϕ_H) and the corresponding sigma points s_1, \dots, s_5 where s_1 denotes the mean value. Figure 12 shows the two-dimensional distribution of the intersection coordinates (x_c, y_c) with the sigma points S_1, \dots, S_5 . S_1 denotes the mean value and S_1, \dots, S_5 are distributed in such a way that the s_i are transformed into $S_i, i = 1 \dots 5$. From both figures, an optimal selection of both s_1, \dots, s_5 and S_1, \dots, S_5 can be observed, which results in a good match of the computed and measured standard deviations σ_{x_c} .

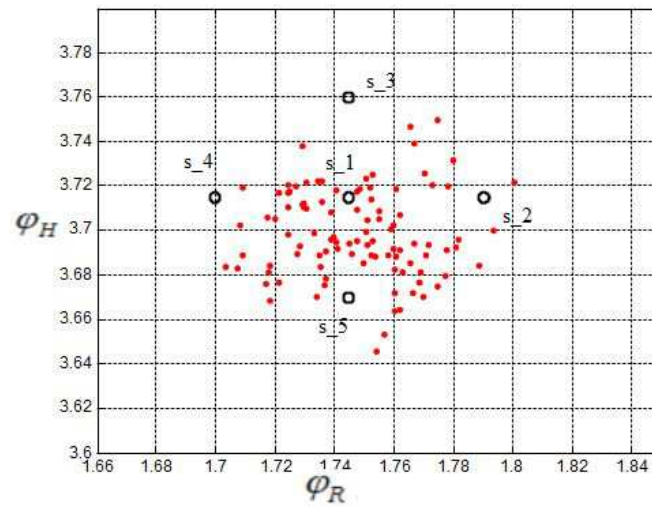


Figure 11. Sigma points, input, units of ϕ_R and ϕ_H in rad.

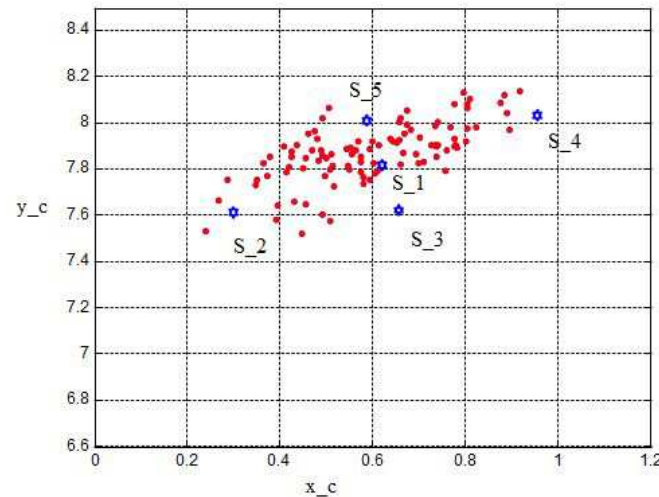


Figure 12. Sigma points, output.

Six-inputs–two-outputs:

The 6-inputs–2-outputs example shows that the additional consideration of 4 input position coordinates with $\sigma_{x_R} = 0.02$ leads to similar results both for the computed and measured covariances and between the sigma-point method and statistical linearization (see $P(7,7) = \sigma_{x_c}^2$, $P(8,8) = \sigma_{y_c}^2$ and $covar(7,7) = \sigma_{x_m}^2$, $covar(8,8) = \sigma_{y_m}^2$, and $\sigma_{x_c}^2$ —computed, and $\sigma_{x_m}^2$ —the measured variation). Table 2 shows the covariance submatrix considering the output positions only.

Computed covariance:

$$P = 10^{-1} \times \begin{pmatrix} 0.004 & -0.000 & -0.000 & 0.000 & -0.000 & -0.000 & -0.030 & -0.018 \\ -0.000 & 0.004 & 0.000 & -0.000 & -0.000 & -0.000 & 0.003 & -0.017 \\ -0.000 & 0.000 & 0.004 & 0.000 & -0.000 & -0.000 & 0.004 & 0.002 \\ 0.000 & -0.000 & 0.000 & 0.004 & -0.000 & -0.000 & 0.001 & 0.000 \\ -0.000 & -0.000 & -0.000 & -0.000 & 0.004 & 0.000 & 0.000 & -0.002 \\ -0.000 & -0.000 & -0.000 & -0.000 & 0.000 & 0.004 & -0.001 & 0.004 \\ -0.030 & 0.003 & 0.004 & 0.001 & 0.000 & -0.001 & \mathbf{0.235} & \mathbf{0.127} \\ -0.018 & -0.017 & 0.002 & 0.000 & -0.002 & 0.004 & \mathbf{0.127} & \mathbf{0.165} \end{pmatrix} \quad (60)$$

$\sigma_{x_c} = 0.153, \quad \sigma_{y_c} = 0.122$

Measured covariance:

$$covar = 10^{-1} \times \begin{pmatrix} 0.004 & 0.000 & 0.000 & 0.000 & 0.000 & -0.000 & -0.028 & -0.020 \\ 0.000 & 0.004 & 0.000 & 0.001 & 0.000 & -0.000 & 0.000 & -0.020 \\ 0.000 & 0.000 & 0.004 & -0.000 & 0.001 & -0.001 & 0.003 & 0.001 \\ 0.000 & 0.001 & -0.000 & 0.004 & -0.000 & -0.000 & -0.000 & -0.003 \\ 0.000 & 0.000 & 0.001 & -0.000 & 0.005 & -0.000 & -0.001 & -0.006 \\ -0.000 & -0.000 & -0.001 & -0.000 & -0.000 & 0.005 & -0.000 & 0.005 \\ -0.028 & 0.000 & 0.003 & -0.000 & -0.001 & -0.000 & \mathbf{0.213} & \mathbf{0.131} \\ -0.020 & -0.020 & 0.001 & -0.003 & -0.006 & 0.005 & \mathbf{0.131} & \mathbf{0.182} \end{pmatrix} \quad (61)$$

$\sigma_{x_c} = 0.145, \quad \sigma_{y_c} = 0.134$

Two-inputs–two-outputs, direct and inverse solution

The next example shows the computation of the direct and inverse cases. In the direct case, we obtain again similar values between the computed and measured covariances and, with this, the standard deviations. The results of the inverse solution lead to similar values of the original inputs (orientations $x_1 = \phi_R, x_2 = \phi_H$) (see Table 2). The simulations of the fuzzy versions showed the same similarities and can therefore be left out here.

Table 2. Covariances, standard deviations—computed and measured.

Outputs	Covariance, Computed	Covariance, Measured	σ_{x_c} Comp/Meas	σ_{y_c} Comp/Meas
2 inputs	$P = \begin{pmatrix} 0.0213 & 0.0114 \\ 0.0114 & 0.0159 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0264 & 0.0146 \\ 0.0146 & 0.0166 \end{pmatrix}$	0.145/0.144	0.126/0.134
2 inputs, stat. lin.	-	-	0.144/0.142	0.124/0.129
6 inputs	$P = \begin{pmatrix} 0.0235 & 0.0127 \\ 0.0127 & 0.0165 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0213 & 0.0131 \\ 0.0131 & 0.0182 \end{pmatrix}$	0.135/0.145	0.122/0.134
Direct solution	$P = \begin{pmatrix} 0.0234 & 0.0133 \\ 0.0133 & 0.0151 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0264 & 0.0146 \\ 0.0146 & 0.0166 \end{pmatrix}$	0.152/0.162	0.128/0.128
Inverse solution	$P = 10^{-3} \times \begin{pmatrix} 0.4666 & 0.0522 \\ 0.0522 & 0.4744 \end{pmatrix}$	$covar = 10^{-3} \times \begin{pmatrix} 0.4841 & -0.0190 \\ -0.0190 & 0.396 \end{pmatrix}$	0.0215/0.0220	0.0217/0.0190

Two-inputs–two-outputs, moving robot–human

The next example deals with the robot and human in motion. Figure 13 shows the positions and orientations of the robot and human at selected time steps $t_1 \dots t_5$ and the development of the corresponding intersections x_c .

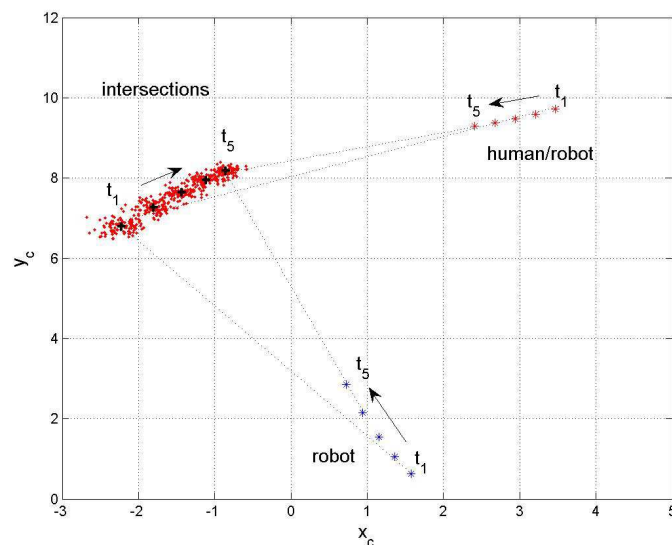


Figure 13. Moving robot and human.

Figure 14 shows the corresponding time plot. The time steps $t_1 \dots t_5$ are taken at 0.58 s, \dots , 4.58 s with a time distance of 1 s, which is 25 time steps of 0.04 s each. The robot and human start at

$$\mathbf{x}_R = (x_R, y_R)^T = (2, 0)^T \text{ m}$$

$$\mathbf{x}_H = (x_H, y_H)^T = (4, 10)^T \text{ m}$$

with the velocities

$$\dot{x}_R(k) = -0.21 \text{ m/s};$$

$$\dot{y}_R(1) = +0.24 \text{ m/s};$$

$$\dot{x}_H(k) = -0.26 \text{ m/s};$$

$$\dot{y}_H(1) = -0.24 \text{ m/s}.$$

k is the time step.

The x components of the velocities $\dot{x}_R(k)$ and $\dot{x}_H(k)$ stay constant during the whole simulation.

The y components change their velocities with constant factors

$$\dot{y}_R(k+1) = K_R \cdot \dot{y}_R(k)$$

$$\dot{y}_H(k+1) = K_H \cdot \dot{y}_H(k)$$

where $K_R = 1.2$ and $K_H = 0.9$. The orientation angles start with the following:

$$\phi_R = 1.78 \text{ rad};$$

$$\phi_H = 3.69 \text{ rad}.$$

They change their values every second according to the direction of motion.

From both plots, one observes an expected decrease in the output standard deviations for a mutual decrease in their distances to the specific intersection and a good match between the computed and measured values \mathbf{x}_c (see Table 3). With the information about the distance of the robot and the standard deviation from and at the expected intersection, respectively, it becomes possible to plan either an avoidance strategy or mutual cooperation between the robot and human.

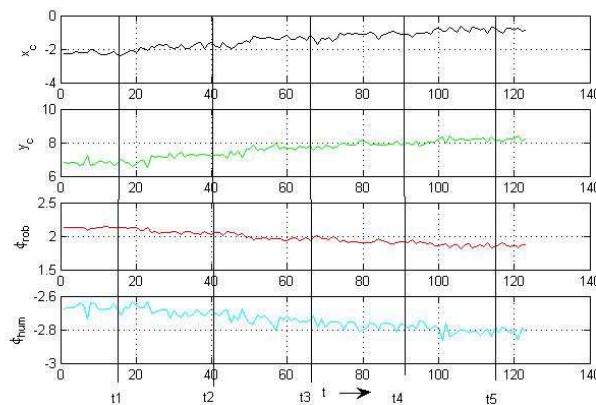


Figure 14. Time plot, robot and human.

Table 3. Covariances, standard deviations—computed and measured, moving robot–human.

Outputs	Covariance, Computed	Covariance, Measured	σ_{x_c} , Comp/Meas	σ_{y_c} , Comp/Meas
t_1	$P = \begin{pmatrix} 0.0220 & 0.0017 \\ 0.0017 & 0.0163 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0246 & -0.0002 \\ -0.0002 & 0.0202 \end{pmatrix}$	0.148/0.156	0.127/0.142
t_2	$P = \begin{pmatrix} 0.0198 & 0.0023 \\ 0.0023 & 0.0138 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0222 & 0.0018 \\ 0.0018 & 0.0153 \end{pmatrix}$	0.140/0.148	0.117/0.123
t_3	$P = \begin{pmatrix} 0.0168 & 0.0030 \\ 0.0030 & 0.0107 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0140 & 0.0040 \\ 0.0040 & 0.0088 \end{pmatrix}$	0.129/0.118	0.103/0.093
t_4	$P = \begin{pmatrix} 0.0151 & 0.0029 \\ 0.0029 & 0.0083 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0127 & 0.0014 \\ 0.0014 & 0.0073 \end{pmatrix}$	0.122/0.112	0.091/0.085
t_5	$P = \begin{pmatrix} 0.0125 & 0.0023 \\ 0.0023 & 0.0061 \end{pmatrix}$	$covar = \begin{pmatrix} 0.0102 & 0.0030 \\ 0.0030 & 0.0056 \end{pmatrix}$	0.112/0.101	0.078/0.074

7. Summary and Conclusions

The content of this work is the prediction of encounter situations of mobile robots and human agents in shared areas by analyzing planned/intended trajectories in the presence of uncertainties and system and observation noise. In this context, the problem of intersections of trajectories with respect to system uncertainties and Gaussian noise of the position and orientation of the agents involved is discussed. The problem is addressed by two methods: the statistical linearization of distributions and the sigma-point transformation of the distribution parameters. The positions and orientations of the robot and human are corrupted with Gaussian noise represented by the parameters' mean and standard deviation. The goal is to calculate the mean and standard deviation/variation at the intersection via the nonlinear relation between the positions/orientations of the robot and human, on the one hand, and the position of the intersection of their intended trajectories, on the other hand.

This analysis is realized by the statistical linearization of the nonlinear relation between the statistics of the robot and human (input) and the statistics of the intersection (output). The output results are the mean and standard deviation of the intersection as functions of the input parameters' mean and standard deviation of the positions and orientations of robot and human. This work is first carried out for two-input–two-output relations (two orientations of the robot–human and two intersection coordinates) and then for six inputs–two outputs (two orientations and four position coordinates of the robot–human and two intersection coordinates). These cases were extended to their fuzzy versions by different Takagi–Sugeno (TS) fuzzy approximations and compared with the non-fuzzy case. Up to a certain resolution, the approximation works as accurately as the original non-fuzzy version. For the two-input–two-output case, an inverse solution is derived, except for the six-input–two-output case because of the undetermined nature of the differential input–output relation.

The sigma-point transformation aims at transforming/propagating distribution parameters—the sigma points—directly through nonlinearities. The transformed sigma points are then converted into the distribution parameters' mean and covariance matrix. The sigma-point transformation is closely connected to the unscented Kalman filter, which is used in the example of the robot and human in motion. The specialty of the example is a computed virtual system output (“observation”)—the intersection of two intended trajectories—where the corresponding output uncertainty is a sum of the transformed position/orientation noise and the computational uncertainty from the fuzzy approximation. In total, the comparison between the computed and measured covariances shows a very good match and the comparison with the statistical linearization shows good coincidences as well. Both the sigma-point transformation and the differential statistical linearization scales for more than two variables linearly. Their computational complexity is in the same order [13]. However, if the model is nonlinear, then the differential linearization (EKF) serves as the first-order or second-order approximating estimator. If the system is highly nonlinear, the EKF may diverge and the sigma-point approach produces typically better results. In summary, a prediction of the accuracy of human–robot trajectories using the methods presented in this work increases the performance of human–robot collaboration and human safety. In future work, this method can be used for robot–human scenarios in factory workshops and for robots working in complicated environments like rescue operations in cooperation with human operators.

Author Contributions: R.P. developed the methods and implemented the simulation programs. A.J.L. conceived the project and gave advice and support. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017274 (DARKO).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MI, USA, 25–28 March 1985; p. 500505.
2. Firl, J. Probabilistic Maneuver Recognition in Traffic Scenarios. Ph.D. Thesis, KIT Karlsruhe Institute of Technology, Karlsruhe, Germany, 2014.
3. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Zhao, X.; Kim, T. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [[CrossRef](#)]
4. Chen, J.; Wang, C.; Chou, C. Multiple target tracking in occlusion area with interacting object models in urban environments. *Robot. Auton. Syst.* **2018**, *103*, 68–82. [[CrossRef](#)]
5. Kassner, M.; Patera, W.; Bulling, A. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, DC, USA, 13–17 September 2014; pp. 1151–1160.
6. Bruce, J.; Wawer, J.; Vaughan, R. Human-robot rendezvous by co-operative trajectory signals. In Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction Workshop on Human-Robot Teaming, Portland, OR, USA, 2–5 March 2015; pp. 1–2.
7. Kunwar, F.; Benhabib, B. Advanced predictive guidance navigation for mobile robots: A novel strategy for rendezvous in dynamic settings. *Intern. J. Smart Sens. Intell. Syst.* **2008**, *1*, 858–890. [[CrossRef](#)]
8. Palm, R.; Lilienthal, A. Fuzzy logic and control in human-robot systems: Geometrical and kinematic considerations. In Proceedings of the WCCI 2018: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 827–834.
9. Palm, R.; Driankov, D. Fuzzy inputs. In *Fuzzy Sets and Systems—Special Issue on Modern Fuzzy Control*; IEEE: Piscataway, NJ, USA, 1994; pp. 315–335.
10. Foulloy, L.; Galichet, S. Fuzzy control with fuzzy inputs. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 437–449. [[CrossRef](#)]
11. Banelli, P. Non-linear transformations of gaussians and gaussian-mixtures with implications on estimation and information theory. *IEEE Trans. Inf. Theory* **2013**.
12. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
13. Merwe, R.v.d.; Wan, E.; Julier, S.J. Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation. In Proceedings of the AIAA 2004, Guidance, Navigation and Control Conference, Portland, OR, USA, 28 June–1 July 2004.
14. Terejanu, G.A. *Unscented Kalman Filter Tutorial*; Department of Computer Science and Engineering University at Buffalo: Buffalo, NY, USA, 2011.
15. Odry, Á.; Kecskes, I.; Sarcevic, P.; Vizvari, Z.; Toth, A.; Odry, P. A novel fuzzy-adaptive extended kalman filter for real-time attitude estimation of mobile robots. *Sensors* **2020**, *20*, 803. [[CrossRef](#)] [[PubMed](#)]
16. Song, Y.; Song, Y.; Li, Q. Robust iterated sigma point fastslam algorithm for mobile robot simultaneous localization and mapping. *Chin. J. Mech. Eng.* **2011**, *24*, 693. [[CrossRef](#)]
17. Bittler, J.; Bhounsule, P.A. Hybrid unscented kalman filter: Application to the simplest walker. In Proceedings of the 3rd Modeling, Estimation and Control Conference MECC, Lake Tahoe, NV, USA, 2–5 October 2023.
18. Alireza, S.D.; Shahri, M. Vision based simultaneous localization and mapping using sigma point kalman filter. In Proceedings of the 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Montreal, QC, Canada, 17–18 September 2011.
19. Xue, Z.; Schwartz, H. A comparison of several nonlinear filters for mobile robot pose estimation. In Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation (ICMA), Kagawa, Japan, 4–7 August 2013.
20. Lyu, Y.; Pan, Q.; Lv, J. Unscented transformation-based multi-robot collaborative self-localization and distributed target tracking. *Appl. Sci.* **2019**, *9*, 903. [[CrossRef](#)]
21. Raitoharju, M.; Piche, R. On computational complexity reduction methods for kalman filter extensions. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 2–19. [[CrossRef](#)]
22. Palm, R.; Lilienthal, A. Uncertainty and fuzzy modeling in human-robot navigation. In Proceedings of the 11th Joint International Computer Conference (IJCCI 2019), Vienna, Austria, 17–19 September 2019; pp. 296–305.
23. Palm, R.; Lilienthal, A. Fuzzy geometric approach to collision estimation under gaussian noise in human-robot interaction. In Proceedings of the Computational Intelligence: 11th International Joint Conference, IJCCI 2019, Vienna, Austria, 17–19 September 2019; Springer: Cham, Switzerland; pp. 191–221.

24. Schaefer, J.; Strimmer, K. A shrinkage to large scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol. Biol.* **2005**, *4*, 32. [[CrossRef](#)] [[PubMed](#)]
25. Cao, Y. Learning the Unscented Kalman Filter. 2021. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/18217-learning-the-unscented-kalman-filter> (accessed on 14 May 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.