Università
degli Studi
di Padova

Dipartimento di
Ingegneria
dell'Informazione

Department of Information Engineering

*Master thesis in Control Systems Engineering*

# A Stochastic Hybrid Control Law for a Localization Task based on Active Sensing

*Supervisor*
Prof. Angelo Cenedese
University of Padova

*Co-supervisor*
Prof. Luca Zaccarian
University of Trento

*Master Candidate*
Giacomo De Lazzari

*Student ID*
2027172

*Academic Year*
2022-2023

“Creativity requires the courage to let go of certainties.”

— Erich Fromm

# Abstract

A localization task by a mobile agent is considered, where the sensing and the motion actions are performed exclusively with respect to each other, meaning that sensing the target is not available during the agent movement. To formalize this scenario the system is modeled on the 1D line and a control law with a timer and a logic variable, to allow switching between the operating modes of sensing the target and moving towards it, is designed. A Stochastic Hybrid System in standard form is obtained, satisfying the Stochastic Hybrid Basic conditions, and the overall closed-loop system behavior is then studied.

An extension to the 2D plane is also proposed and its behavior is analyzed. Finally, the 2D control law is tested for a simulated Search & Rescue task in an unknown indoor environment under some simplifying hypotheses.

# Contents

# List of figures

# List of tables

# List of acronyms

| | |
|---|---|
| **AS** | Active Sensing |
| **DOA** | Direction Of Arrival |
| **MLE** | Maximum Likelihood Estimate |
| **MVE** | Minimum Variance Estimate |
| **p.d.f.** | probability density function |
| **RBE** | Recursive Bayesian Estimation |
| **SAR** | Search And Rescue |
| **SHS** | Stochastic Hybrid System |

# 1 Introduction

The work of this thesis stems from the search for applications of Hybrid Systems theory in Active Sensing problems. This seems to be an unexplored combination in the literature, and the aim of the work is to explore possible results at the intersection of these two fields.

## 1.1 Active Sensing

### 1.1.1 Overview

Active Sensing (AS) involves the control of a system, possessing both actuation and sensing parts, in which control decisions about the sensing are also needed. We may say that, in the AS framework, the control of aspects related to the system sensing is as central as the control of the actuators. An interplay between estimation, perception and control arises. This makes AS highly relevant in automation and robotics tasks in the literature. It is employed, for instance, in Search & Rescue missions, environmental mapping and exploration, path planning, optimal sensor coverage [1].

### 1.1.2 Localization tasks

In the Active Sensing for Search and Tracking review by Luca Varotto et al. [1], three main types of tasks are distinguished: Active Search, Active Localization and Active Tracking.

Active Search involves making data-collection decisions in order to find objects of interest (*targets*) which have not been detected yet. In Active Localization, the purpose is to estimate the location of static targets that have already been detected [2] [3] [4]. Finally, Active Tracking involves non-stationary targets, which require a continuous update of the changing position estimate.

In this context, the task we deal with in the thesis is an Active Localization task. We in fact assume the target to be present, stationary, and only care about finding its location and moving towards it.

## 1.2 Stochastic Hybrid Systems

### 1.2.1 Hybrid Systems

Consider the distinction between continuous and discrete time dynamical systems. Those two classes are well suited for the modeling and study of distinct problems. Sometimes, however, there are dynamical systems whose description can benefit both of continuous-time laws, and of discontinuous transitions of the state. As an example, consider electronic circuits where switching quantities are of core importance to their behavior (e.g. power electronics), or mechanical systems such as a bouncing ball that (with a simplified model) instantly inverts the direction of its velocity vector on collision with the ground [5].

Hybrid Systems allow the description of such dynamical systems, and the field arising from their study provides very useful tools for understanding their behavior and properties. In particular, various stability concepts have been defined and Lyapunov-based results, alongside invariance principles, are available [5].

### 1.2.2 Stochasticity in Hybrid Systems

Stochastic Hybrid Systems (SHSs) also allow for stochastic behavior. For instance, a diffusion process might replace the deterministic flow, the jumps might happen at random instants, or the jump map may depend on some random quantity. Not all results available for deterministic Hybrid Systems have currently been obtained for SHSs [6], however the literature provides useful results for specific classes of such systems, for instance [7]. The 2014 survey from Teel et al. [6] provides an overview of the current results.

## 1.3 Audio sensing and selected task

The first step involved seeking a suitable task to explore the intersection between AS and Hybrid Systems.

The idea originated from the application of sound sensing to localization tasks. An example scenario could be a Search & Rescue mission in a devastated building, where the goal is to localize and rescue the victims (by driving towards them) with the aid of a sound sensing device. On first sight, this is an interesting idea, since other sensing modalities might suffer in such environments. Consider, for instance, visual sensing: if there is a lot of dust and/or smoke, the camera might not detect a victim in the room. Moreover, if there is no direct sight between the autonomous agent, no information can be provided by such type of sensor.

There is already various work in the literature involving Search and Localization tasks with the aid of audio sensing, for instance [8] [9] [10] [11] [12], together with various work on design and improvement of sound source Direction Of Arrival (DOA) estimation through microphone arrays [13] [14].

With audio sensing, it might also be possible to drive towards a victim in an unknown environment even when the agent and victim are not in direct sight, due to the sound waves propagating in corridors and through doors and walls. This is better discussed towards the end in chapter 4.

However, another peculiarity of this sensing modality, applied to mechanical autonomous agents, is that the agent actuation disturbs the sound readings. Previous work also references this issue in the literature [8] [9] [15] [16].

There have been attempts to process the audio signals and remove the agent's self-generated noise [15], and to combine audio, visual and motion information [16] for a more accurate estimate. One might also argue that improvements in mechanical actuators might mitigate this issue but, for instance, in the hypothesis of a Search & Rescue task in a devastated building, we observe that the sounds produced by the interaction of the agent actuation and the environment (like driving or walking over debris and obstacles) cannot be entirely controlled.

Then the solution explored in this work is based on the pessimistic assumption that the mechanical noise produced by the actuation completely invalidates the measurements. This lays the ground for an interesting problem to tackle with Hybrid Systems tools, which have already proven useful for the design of switching control laws for tasks requiring online optimization decisions [17] [18].

In this work, we will not deal with the challenges related to the design of sound source DOA estimators, but will instead assume that noisy DOA readings are available (with an appropriate simplified model) when the agent is stationary.

We abstract the problem by moving it from the specific modality of audio sensing and obtain a description which considers noisy position readings to only be available when the agent is not moving. Those could also originate from electromagnetic sensing or other physical phenomena, but we preserve the important assumption of exclusivity between the two actions of measuring and moving. We might then extend the ideas developed in this work to other tasks where this tradeoff is present.

# 2

# One-dimensional task

## 2.1 Description and modeling

In this section, we describe the first simplified scenario that was considered for the thesis work. It involves a single autonomous agent tasked with localizing a target (which we might think of as the victim in a Search & Rescue task) whose position is unknown but stationary. Moreover, the measurements are noisy and are unavailable while the agent is moving.

The fundamental assumptions for the problem are then the following:

**Assumption 1 (A1):** *The task happens in $\mathbb{R}^1$: the agent and the target are located on a one-dimensional space and their positions in time are $x_A(t) \in \mathbb{R}$ and $x_T(t) \in \mathbb{R}$ respectively.*

**Assumption 2 (A2):** *The agent movement is controlled through the velocity input signal $u(t)$ $[\mathrm{m/s}]$, so we have $\dot{x}_A(t) = u(t)$.*

**Assumption 3 (A3):** *The target is assumed to be stationary, that is $\dot{x}_T(t) \equiv 0\ \forall t$.*

**Assumption 4 (A4):** *The distance sensing unit of the agent doesn't work when $\dot{x}_A(t) \neq 0$, that is when the agent is moving.*

**Assumption 5 (A5):** *A distance measure requires $T_m$ time to be ready, during which the condition of A4 must hold.*

**Assumption 6 (A6):** *Distance measurements provided by the agent's distance sensing unit are noisy, and more specifically, the additive noise is normally distributed with variance $\sigma_d^2 > 0$. The additive Gaussian errors are independent from each other.*

The objective is for the autonomous agent to drive towards the target, subject to the constraints outlined by the assumptions above.

### 2.1.1 Task setup in $\mathbb{R}^1$

Following the description above, and starting from A1, we define the distance between the agent and the target as:

$$d(t) := x_T(t) - x_A(t) \tag{2.1}$$

Considering A2 and A3, we obtain the following dynamical equation for the agent-target distance:

$$\dot{d}(t) = \dot{x}_T(t) - \dot{x}_A(t) \tag{2.2}$$

$$= -u(t) \tag{2.3}$$

which makes it clear how the agent can minimize $d(t)$, thus driving towards $x_T(t)$, with the control signal $u(t)$ (assuming $d(t)$ is known).

However, the distance is not known, but as we outlined above, it can be measured through the sensing unit. We denote a measurement taken at time $t$ as $\tilde{d}(t)$, and we consider A6 to write the following:

$$\tilde{d}(t) = d(t) + e(t) \qquad \text{where } e(t) \sim \mathcal{N}(0, \sigma_d^2), \quad e(t_1) \perp e(t_2) \quad \forall t_1 \neq t_2 \tag{2.4}$$

### 2.1.2 Switched mode control

A4 and A5 suggest that the controller must choose between two different incompatible behaviors:

1. moving the agent with $u(t) \neq 0$, physically progressing towards the target, preventing however the availability of distance measurements;
2. taking measurements, thus acquiring information about the target position, but without moving and thus physically progressing towards it.

It is critical to notice that both behaviors are necessary for the task completion: it is true that the agent can complete the task by driving towards the target (1.), however this can only happen if the agent has some information about its location, which can only be obtained in *mode* (2.).

We then introduce the two following operating modes for the controller, associated with a logical variable that we define as $q(t) \in \{0, 1\}$ and will be part of the controller state:

- $q = 0$: Measurement mode
- $q = 1$: Movement mode

The main design areas for the controller will be the switching conditions for the two modes, and the feedback law for $u(t)$.

### 2.1.3 Timed measurements

Before working on the switching conditions and the agent velocity input control law, we focus on modeling the timing of the measurements and the fusion of the information they provide.

A5 requires that $T_m$ time elapses, during which it must be $u(t) \equiv 0$, for a measurement to be available. We then introduce a timer state variable in the controller so that it can keep track of this required time period while the measurement mode is active. Let this variable be

$$\tau(t) \in [0, T_m] \subset \mathbb{R} \tag{2.5}$$

which is defined on a compact subset of $\mathbb{R}$. We assume that the timer increments as

$$\dot{\tau}(t) = 1 \tag{2.6}$$

when in measurement mode, and we will design the hybrid system so that $\tau(t) = T_m$ leads to a jump that will both

- reset the timer to $\tau^+(t) = 0$,
- consider the measurement $\tilde{d}(t)$ from the sensor.

### 2.1.4 Information fusion

We now formalize the way we perform information fusion with the RBE approach to keep an on-line estimate of the agent-target distance.

We will differentiate between measurement and movement mode, since the assumptions about the available information and underlying dynamics are different: in the first case, the target and the agent (and consequently their distance) are stationary, with measurements being available. In the second case, instead, measurements are not available and the distance is changing due to agent movement.

At the end of this section, it will be clear that the scheme we adopt is a Kalman filtering one [19], and the two operating modes of the control law will exclusively leverage either the prediction or the update mechanism of this technique.

#### While in measurement mode

Let us assume that the agent is in measurement mode for a given time interval, during which distance measurements happen. As described in the previous section, these measurements will be equally spaced in time with period $T_m$. Let us define

$$\{t_i \in \mathbb{R}^+\}_{i=1,\dots,N}, \quad t_{i+1} = t_i + T_m \;\; \forall i < N \tag{2.7}$$

for some integer $N$ number of measurements. The ordered times $t_i$ denote the instants at which measurements happen.

Following A6, we define the p.d.f. of the measurement $\tilde{d}_{t_i} = \tilde{d}(t_i)$ at the time $t_i$ as:

$$p(\tilde{d}_{t_i} \mid d_{t_i}) = \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\frac{(\tilde{d}_{t_i} - d_{t_i})^2}{\sigma_d^2}\right) \tag{2.8}$$

which, in particular, depends on the current state of the system $d_{t_i}$. The dynamical model of the agent-target distance in the measurement phase ($\dot{d}(t) = 0$) leads to the distance $d(t)$ being constant, and the discrete-time process $\{d_{t_i}\}$ obtained by repeated sampling, being Markov. That is

$$p(d_{t_i} \mid d_{t_{i-1}}, \ldots, d_{t_0}) = p(d_{t_i} \mid d_{t_{i-1}}) \tag{2.9}$$

in particular, since there is no uncertainty in the actuation, and no process noise in the target dynamic, we can write

$$p(d_{t_i} \mid d_{t_{i-1}}) = \delta(d_{t_i} - d_{t_{i-1}}) \tag{2.10}$$

where $\delta(\cdot)$ is the measurable Dirac delta generalized function.

We now proceed with the standard reasoning for implementing RBE, for instance, as in [20].

We are looking for $p(d_{t_i} \mid \tilde{d}_{t_i}, \ldots, \tilde{d}_{t_0})$, which represents the p.d.f. of the agent-target distance given the measurements up to the current time $t_i$. We expand the expression as follows:

$$\begin{aligned}
p(d_{t_i} \mid \tilde{d}_{t_i}, \ldots, \tilde{d}_{t_0}) &= p(d_{t_i} \mid \tilde{d}_{t_i}, \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) && (2.11)\\
&= \frac{p(\tilde{d}_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}, d_{t_i}) p(d_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0})}{p(\tilde{d}_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0})} && (2.12)\\
&= \frac{p(\tilde{d}_{t_i} \mid d_{t_i}) p(d_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0})}{p(\tilde{d}_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0})} && (2.13)\\
&\propto p(\tilde{d}_{t_i} \mid d_{t_i}) p(d_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) && (2.14)
\end{aligned}$$

where, in the last expression (2.14), we have the product of the likelihood and the prior.

Moreover, using the dynamical prior of Equation (2.10) in the Chapman-Kolmogorov equation applied to Markov chains [21, chapter 2.2]:

$$\begin{aligned}
p(d_{t_i} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) &= \int p(d_{t_i} \mid d_{t_{i-1}}) p(d_{t_{i-1}} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) \, \mathrm{d}d_{t_{i-1}} && (2.15)\\
&= \int \delta(d_{t_i} - d_{t_{i-1}}) p(d_{t_{i-1}} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) \, \mathrm{d}d_{t_{i-1}} && (2.16)\\
&= p(d_{t_{i-1}} \mid \tilde{d}_{t_{i-1}}, \ldots, \tilde{d}_{t_0}) \Big|_{d_{t_{i-1}} = d_{t_i}} && (2.17)
\end{aligned}$$

by the sampling property of the Dirac delta generalized function.

Let us assume that, initially, due to absence of any information, the p.d.f. of the estimate $d_{t_0}$ at time $t_0$, with the first measurement $\tilde{d}_{t_0}$ being available, is the same as the p.d.f. of the measurement

$$\begin{aligned}
p(d_{t_0} \mid \tilde{d}_{t_0}) &:= p(\tilde{d}_{t_0} \mid d_{t_0}) && (2.18)\\
&= \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\frac{(\tilde{d}_{t_0} - d_{t_0})^2}{\sigma_d^2}\right) && (2.19)\\
&= \frac{1}{\sqrt{2\pi P_{t_0}}} \exp\left(-\frac{1}{2}\frac{(\mu_{t_0} - d_{t_0})^2}{P_{t_0}}\right) && (2.20)
\end{aligned}$$

and is thus a Gaussian p.d.f., where we defined

$$\mu_{t_0} := \tilde{d}_{t_0} \qquad\qquad P_{t_0} := \sigma_d^2 \tag{2.21}$$

to highlight that they characterize the initial estimate at time $t_0$. The prediction step with Equation (2.17) leads to

$$\begin{aligned} p(d_{t_1} \mid \tilde{d}_{t_0}) &= p(d_{t_0} \mid \tilde{d}_{t_0})\Big|_{d_{t_0}=d_{t_1}} && (2.22)\\ &= \frac{1}{\sqrt{2\pi P_{t_0}}} \exp\left(-\frac{1}{2}\frac{(\mu_{t_0} - d_{t_0})^2}{P_{t_0}}\right)\Big|_{d_{t_0}=d_{t_1}} && (2.23)\\ &= \frac{1}{\sqrt{2\pi P_{t_0}}} \exp\left(-\frac{1}{2}\frac{(\mu_{t_0} - d_{t_1})^2}{P_{t_0}}\right) && (2.24) \end{aligned}$$

which is still a Gaussian p.d.f. with mean $\mu_{t_0}$ and variance $P_{t_0}$, as expected due to the constant dynamic. The update step, which happens with the new measurement $\tilde{d}_{t_1}$ being available, leads to

$$\begin{aligned} p(d_{t_1} \mid \tilde{d}_{t_1}, \tilde{d}_{t_0}) &\propto p(\tilde{d}_{t_1} \mid d_{t_1})p(d_{t_1} \mid \tilde{d}_{t_0}) && (2.25)\\ &= \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\frac{(\tilde{d}_{t_1} - d_{t_1})^2}{\sigma_d^2}\right) \frac{1}{\sqrt{2\pi P_{t_0}}} \exp\left(-\frac{1}{2}\frac{(\mu_{t_0} - d_{t_1})^2}{P_{t_0}}\right) && (2.26) \end{aligned}$$

that is, a p.d.f. proportional to the product of two Gaussian p.d.f. of means $\mu_{t_0}$ and $\tilde{d}_{t_1}$, and variances $P_{t_0}$ and $\sigma_d^2$, respectively. It is a known result that such a product is a p.d.f. proportional to a Gaussian one, and the normalization factor at the denominator that we neglected in the last step of Equation (2.14) leads to the new posterior estimate at time $t_1$ being a Gaussian p.d.f.:

$$p(d_{t_1} \mid \tilde{d}_{t_1}, \tilde{d}_{t_0}) = \frac{1}{\sqrt{2\pi P_{t_1}}} \exp\left(-\frac{1}{2}\frac{(\mu_{t_1} - d_{t_1})^2}{P_{t_1}}\right) \tag{2.27}$$

with

$$\mu_{t_1} = \frac{P_{t_0}\tilde{d}_{t_1} + \sigma_d^2\mu_{t_0}}{P_{t_0} + \sigma_d^2} \qquad\qquad P_{t_1} = \frac{P_{t_0}\sigma_d^2}{P_{t_0} + \sigma_d^2} \tag{2.28}$$

Iteratively it should be intuitive to understand that Equation (2.28) is the update equation for the information fusion that happens each time a new measurement arrives. We can generalize it for any generic update time $t_i$ as

$$\mu_{t_i} = \frac{P_{t_{i-1}}\tilde{d}_{t_i} + \sigma_d^2\mu_{t_{i-1}}}{P_{t_{i-1}} + \sigma_d^2} \qquad\qquad P_{t_i} = \frac{P_{t_{i-1}}\sigma_d^2}{P_{t_{i-1}} + \sigma_d^2} \tag{2.29}$$

and, in particular, we can rewrite it as

$$\mu_{t_i} = K(P_{t_{i-1}})\tilde{d}_{t_i} + (1 - K(P_{t_{i-1}}))\mu_{t_{i-1}} \qquad\qquad P_{t_i} = K(P_{t_{i-1}})\sigma_d^2 \tag{2.30}$$

where we define the Kalman gain

$$K(P) := \frac{P}{P + \sigma_d^2} \tag{2.31}$$

It is then clear that the estimate of the agent-target distance at some specific time in the measurement phase is represented by a Gaussian random variable. The parameters that characterize this random variable can be updated on measurements with Equation (2.30). It follows naturally that we introduce the two state variables

$$\mu(t) \in \mathbb{R} \tag{2.32}$$
$$P(t) \in \mathbb{R}^+ \tag{2.33}$$

to encode the current normally distributed estimate of $d(t)$.

## While in movement mode

In the movement phase, the measurements are not available. Instead, we have

$$\dot{x}_A(t) = u(t) \neq 0 \tag{2.34}$$
$$\implies \dot{d}(t) \neq 0 \tag{2.35}$$

Here, the dynamical model differs from before, and Equation (2.10) doesn't hold anymore. Moreover, the dynamic is now continuous in time, in contrast to the previous case of the measurements happening instantaneously in some specific time instants.

Instead, we may introduce $\mathrm{dt} > 0$ very small and write:

$$p(d_{t+\mathrm{dt}} \,|\, d_t) = \delta(d_{t+\mathrm{dt}} - (d_t + \dot{d}_t \, \mathrm{dt})) \tag{2.36}$$

We then recall the p.d.f. of the estimate of $d(t) = d_t$ at time $t$:

$$p(d_t) = \frac{1}{\sqrt{2\pi P_t}} \exp\left(-\frac{1}{2}\frac{(d_t - \mu_t)^2}{P_t}\right) \tag{2.37}$$

and expand on $p(d_{t+\mathrm{dt}})$ as follows:

$$\begin{aligned}
p(d_{t+\mathrm{dt}}) &= \int p(d_{t+\mathrm{dt}} \,|\, d_t) p(d_t) \,\mathrm{d}d_t && (2.38)\\
&= \int \delta(d_{t+\mathrm{dt}} - (d_t + \dot{d}_t \,\mathrm{dt})) p(d_t) \,\mathrm{d}d_t && (2.39)\\
&= \int \delta((d_{t+\mathrm{dt}} - \dot{d}_t \,\mathrm{dt}) - d_t) p(d_t) \,\mathrm{d}d_t && (2.40)\\
&= p(d_t) \,\mathrm{d}d_t \Big|_{d_t = d_{t+\mathrm{dt}} - \dot{d}_t \,\mathrm{dt}} && (2.41)\\
&= \frac{1}{\sqrt{2\pi P_t}} \exp\left( -\frac{1}{2} \frac{(d_{t+\mathrm{dt}} - \dot{d}_t \,\mathrm{dt} - \mu_t)^2}{P_t} \right) && (2.42)\\
&= \frac{1}{\sqrt{2\pi P_t}} \exp\left( -\frac{1}{2} \frac{(d_{t+\mathrm{dt}} - (\mu_t + \dot{d}_t \,\mathrm{dt}))^2}{P_t} \right) && (2.43)
\end{aligned}$$

We then conclude that the new estimate at time $t + \mathrm{dt}$ is represented by

$$\begin{aligned}
P_{t+\mathrm{dt}} &= P_t && (2.44)\\
\mu_{t+\mathrm{dt}} &= \mu_t + \dot{d}_t \,\mathrm{dt} && (2.45)
\end{aligned}$$

In particular Equation (2.45) implies that

$$\mu(t + \mathrm{dt}) - \mu(t) = \dot{d}(t) \,\mathrm{dt} \,. \tag{2.46}$$

and, dividing by dt and taking the limit for $\mathrm{dt} \to 0$ of both sides of this last equation, we obtain

$$\begin{aligned}
\lim_{\mathrm{dt} \to 0} \frac{\mu(t + \mathrm{dt}) - \mu(t)}{\mathrm{dt}} &= \lim_{\mathrm{dt} \to 0} \frac{\dot{d}(t) \cancel{\mathrm{dt}}}{\cancel{\mathrm{dt}}} && (2.47)\\
\dot{\mu}(t) &= \dot{d}(t) && (2.48)
\end{aligned}$$

We then recall the equation for $\dot{d}(t)$ from the task setup described in subsection 2.1.1 and obtain the continuous-time RBE prediction equation for $\mu(t)$ to be used in movement mode:

$$\dot{\mu}(t) = -u(t) \tag{2.49}$$

together with the following prediction equation for $P(t)$, obtained with the same procedure from Equation (2.44):

$$\dot{P}(t) = 0 \tag{2.50}$$

This result is compatible with the one obtained by continuous-time Kalman filtering [19].

### Observations

- Due to the absence of process noise, as long as the agent controller is in measurement mode ($q = 0$) the variance $P(t)$ of the distance estimate will be monotonically decreasing and become arbitrarily small as $t \to \infty$. This is because in Equation (2.30) the variance $P_{t_i}$ updates through the map

  $$P \longmapsto (1 - K(P))P$$

  where

  $$1 - K(P) = \frac{\sigma_d^2}{P + \sigma_d^2} < 1$$

  since $\sigma_d^2 > 0$ and $P \geq 0$. The map is then a contraction with $P^* = 0$ as fixed point.
- When the agent controller is instead in movement mode ($q = 1$), the variance will stay constant.
- We will use $\mu(t)$ as the "best" estimate value of $d(t)$, which is the result we obtain with both the maximum-likelihood and the minimum variance criteria.

## 2.1.5 Control input and movement action

We now proceed with the design of a feedback law for the movement phase. The goal is to drive the distance $d(t)$ towards zero.

Recall that $\dot{d}(t) = -u(t)$. We might be tempted to choose a feedback law such as

$$u(t) = \alpha d(t) \tag{2.51}$$

with $\alpha > 0$ which would guarantee exponential convergence $d(t) \to 0$.

However, this is not possible, since the quantity $d(t)$ is not known by the agent. In its place, we instead use the mean of its estimate p.d.f., which is part of the state of the RBE subsystem we designed in the previous subsection, following the last observation we made about it being the MVE and MLE:

$$u(t) = \alpha\mu(t) \tag{2.52}$$

In this way, we are closing the loop with the value provided by the estimator.

## 2.1.6 Mode switching conditions and reset

Finally, we are left with the design of the mode switching conditions: like we described in subsection 2.1.2, it is necessary for the agent to make use of both operating modes. We may say that the purpose of the measurement mode is to perform distance measurements to obtain an estimate of the agent-target distance. This estimate is of critical importance since it drives the movement in the homonymous operating mode. It then makes sense that the controller will stay in measurement mode until the estimate is deemed "good enough". Of course, we must define what we mean by "good enough".

The first thing that comes to mind is that the estimate variance $P(t)$ is monotonically decreasing during the measurement phase and is inversely proportional to the quality of the estimate. It seems reasonable to design the measurement-to-movement mode switch so that it happens when the estimate variance $P(t)$ goes below a specific threshold.

Conversely, in movement mode, it seems reasonable to keep moving while the (estimated[1]) distance to the target is above a given threshold, being the feedback law in Equation (2.52) designed to make the distance strictly monotonically decreasing. We then expect to end the movement mode when $\mu(t)$ goes below a specific threshold.

To design the thresholds, we start with the observation that $\sqrt{P(t)}$ is the standard deviation of the Gaussian estimate. We can multiply this quantity by some positive scalar coefficient and obtain a value in the *meter* SI physical unit. Moreover, such quantity represents a confidence interval around the estimate of the agent-target distance. It then made sense to choose the movement-to-measurement switching condition as

$$|\mu(t)| \leq b\sqrt{P(t)} \tag{2.53}$$

where $b \in \mathbb{R}^+$ is a parameter of the control law. In this way, the comparison is physically meaningful (both quantities are in meters) and we obtain an interesting behavior: the more accurate the estimate of $d(t)$ is, the closer we drive before we stop the movement with respect to the mean of the distance estimate $\mu(t)$.

With reference to the switching condition we just found, and the previous discussion regarding the idea of choosing a threshold on $P(t)$ so that we wait for it to be "small enough", we may propose the following *specular* condition for the measurement-to-movement mode switch:

$$a\sqrt{P(t)} \leq |\mu(t)| \tag{2.54}$$

with $a \in \mathbb{R}^+$ being another parameter for the controller. As before, this leads to an interesting behavior: the closer we are to the target, the lower the threshold we require on the variance $P(t)$, or equivalently, the higher we require the estimate accuracy to be.

To ensure there is some hysteresis between the mode switches, and there is no possibility of overlap in the conditions, we start by noticing that the opposite inequalities (that thus imply that the controller does not perform any mode switch and keeps running, instead, in the current mode), are

$$\begin{cases} |\mu(t)| \geq b\sqrt{P(t)} \\ a\sqrt{P(t)} \geq |\mu(t)| \end{cases} \tag{2.55}$$

[1] Recall that the true distance $d(t)$ is not available to the agent, and the reasonable thing we do is consider the mean of its estimate $\mu(t)$ in its place.

and we combine both expressions to highlight that it must be[2]

$$a\sqrt{P(t)} \geq |\mu(t)| \geq b\sqrt{P(t)} \tag{2.56}$$

$$a \geq \frac{|\mu(t)|}{\sqrt{P(t)}} \geq b \tag{2.57}$$

and so, for there to be a region in the space of $\mu$, $P$ of continuous operation without mode switches, we must have

$$a > b \tag{2.58}$$

and, finally, the conditions will be expressed by taking the square of both sides of (2.53) and (2.54) as follows:

$$\mu^2(t) \leq b^2 P(t) \qquad\qquad a^2 P(t) \leq \mu^2(t) \tag{2.59}$$

Moreover, we design the system so that a movement-to-measurement mode switch leads to a *reset* of the RBE subsystem. This is a key feature of the control scheme, since it allows the system to recover from bad estimates. Those can happen especially if the initial state of the estimator is not meaningful with respect to the true distance. Consider, as an example, the case of

$$d(0) = 10\text{m} \qquad\qquad \mu(0) = 2\text{m} \qquad\qquad P(0) = 0.01 \tag{2.60}$$

which encodes a confidence region of $\pm 0.3$m at $0.997$ probability, with $d(0)$ being way outside of such interval. It is not difficult to see that the switched mode controller, assuming a scenario with $\sigma_d^2 \gg P(0)$, cannot correct for the bad initial state during the measurement phase, due to the Kalman gain (Equation (2.31)) being small. Moreover, it will switch quite soon to the movement phase (this depends on the choice of the parameter $a$, of course) with $\mu(t_s) \sim \mu(0) = 2 \nsim \mathrm{d}(t_s) = d(0)$ at the time $t_s$ of the switch and consequently it will not converge close to $d(\infty) = 0$, but instead at $\mathrm{d}(\infty) \sim 8 = d(0) - \mu(0)$. This behavior can be observed in Figure 2.4b where the final closed-loop system is simulated with a similar initial condition (the initial variance $P(0)$ is not as small, but it is small enough that the described behavior is observed).

As we hinted at the beginning, the chosen solution is to perform a reset of the estimator state with the movement-to-measurement mode switch. The details of how this is done in the final standard form SHS will be discussed later. For the time being, it can be intuitively explained by setting[3] the variance of the estimate $P(t_r^+) = \infty$ after the mode switch and reset jump at time $t_r$. With such a choice, the next measurement update at time $t_{i_0} = t_r + T_m$, with still $P(t_{i_0}) = P(t_r^+) = \infty$, will lead to

$$\lim_{P(t_{i_0})\to\infty} K(P(t_{i_0})) = \lim_{P(t_{i_0})\to\infty} \frac{P(t_{i_0})}{P(t_{i_0}) + \sigma_d^2} = 1 \tag{2.61}$$

[2] In the manipulations that followed, we assumed $P(t) > 0$, but the reasoning can be extended by continuity by taking the limit for $P(t) \to 0^+$.

[3] We can't of course set the state variable to $\infty$, since it is not part of $\mathbb{R}^+$, but the reasoning is done with this notation abuse for simplicity, and is treated later in a formally correct way.

which allows to obtain the re-initialization of the RBE state with the first measurement data:

$$\mu(t_{i_0}^+) = \lim_{P(t_{i_0})\to\infty} K(P(t_{i_0}))\tilde{d}(t_{i_0}) + (1 - K(P(t_{i_0})))\mu(t_{i_0}) \tag{2.62}$$

$$= \tilde{d}(t_{i_0}) \tag{2.63}$$

$$P(t_{i_0}^+) = \lim_{P(t_{i_0})\to\infty} (1 - K(P(t_{i_0})))P(t_{i_0}) \tag{2.64}$$

$$= \lim_{P(t_{i_0})\to\infty} \frac{\sigma_d^2}{P(t_{i_0}) + \sigma_d^2} P(t_{i_0}) \tag{2.65}$$

$$= \left( \lim_{P(t_{i_0})\to\infty} \frac{P(t_{i_0})}{P(t_{i_0}) + \sigma_d^2} \right) \sigma_d^2 \tag{2.66}$$

$$= \sigma_d^2 \tag{2.67}$$

Finally, we chose to perform the mode switch when the timer $\tau = 0$. This is especially relevant for the measurement-to-movement mode switch and, in fact, allows, for instance, the following sequence of events to happen cleanly and in a well-defined way:

- the system performs a reset;
- after $T_m$ units of time have elapsed, a measurement is taken and, the system being in the reset state, set as the current estimate $\mu$;
- the measurement-to-movement mode switch threshold triggers immediately, and the system switches to movement mode.

### 2.1.7 Closed-loop system

We now describe the final closed-loop SHS by defining its data with the same notation as in [7], and later proving that it satisfies the *Stochastic Hybrid Basic Conditions*. At the end of the section Figure 2.3 shows a block diagram of the system that was described, with each block roughly mapping to each one of the previous sections that discussed the different parts. The diagram of Figure 2.2 illustrates the mode switching behavior of the system with reference to the sets and maps that we will soon introduce.

Let the system state be

$$x = \begin{bmatrix} \tau \\ d \\ \mu \\ E \\ q \end{bmatrix} \in \mathcal{X} = [0, T_m] \times \mathbb{R} \times \mathbb{R} \times [0,1] \times \{0,1\} \subset \mathbb{R}^5 \tag{2.68}$$

with the relation

$$E = e^{-1/P} \in [0,1] \qquad\qquad P = \frac{1}{\ln(E^{-1})} \in [0, +\infty] \tag{2.69}$$

Table 2.1 briefly describes all the state variables and lists their physical unit.

We will use $E$ and $P$ interchangeably to refer to the variance of the estimate in its *encoded* vs. original representation, respectively. We imply the use of the invertible change of variable, with the needed extensions by continuity, when we refer to $E$ instead of $P$, and vice versa. Figure 2.1 plots this continuous mapping for a limited range of $P > 0$.

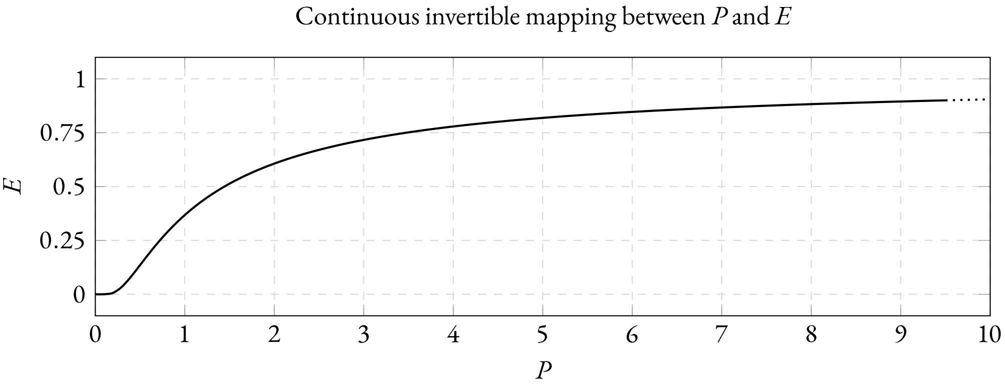


**Figure 2.1:** Plot of the continuous mapping between $P$ and $E$.

Some additional observations about the state variable $E$ are made:

- $E = 1$ encodes the condition of "zero knowledge" ($P = \infty$), and is the way we solve the issue we hinted at before of $P = \infty$ not being a valid state value. In fact, observe that the previous reasoning about the behavior of such value of $P$ in the first measurement update after a reset involved taking the limit for $P \to \infty$, and we have

$$\lim_{P\to\infty} e^{-1/P} = e^0 = 1 \tag{2.70}$$

The reset jump will set $E(t_r^+) = 1$ and the measurement jump map is designed such that

$$E(t) = 1 \implies \begin{cases} \mu(t^+) = \tilde{d}(t) \\ E(t^+) = e^{-1/\sigma_d^2} \end{cases} \tag{2.71}$$

- $E = 0$ encodes the condition of "perfect knowledge" ($P = 0$), in fact

$$\lim_{P\to 0^+} e^{-1/P} = \lim_{z\to\infty} e^{-z} = 0 \qquad \text{and} \qquad \lim_{E\to 0^+} \frac{1}{\ln(E^{-1})} = \lim_{z\to\infty} \frac{1}{\ln(z)} = 0 \tag{2.72}$$

We finally proceed to define the jump and flow sets, the jump maps and the flow vector field of the SHS.

Let us define, in a compact way (which assumes extension of the sets by continuity of the comparison func-

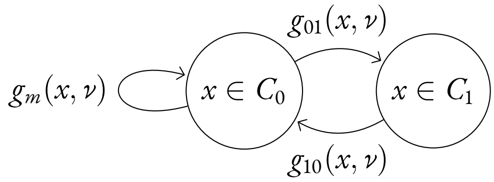


**Figure 2.2:** State transition diagram of the closed-loop stochastic hybrid system, with respect to the two flow sets $C_0$ and $C_1$, tied to the measurement and movement modes, respectively. The transitions coincide with the jumps, which are state-driven, and map the state through the function on the edge label.

tions), the following sets

$$D_m = \{T_m\} \times \mathbb{R} \times \{E \geq e^{-a^2/\mu^2}\} \times \{0\} \tag{2.73}$$

$$C_0 = [0, T_m] \times \mathbb{R} \times \{E \geq e^{-a^2/\mu^2}\} \times \{0\} \qquad D_{01} = \{0\} \times \mathbb{R} \times \{E \leq e^{-a^2/\mu^2}\} \times \{0\} \tag{2.74}$$

$$C_1 = \{0\} \times \mathbb{R} \times \{E \leq e^{-b^2/\mu^2}\} \times \{1\} \qquad D_{10} = \{0\} \times \mathbb{R} \times \{E \geq e^{-b^2/\mu^2}\} \times \{1\} \tag{2.75}$$

with the following meaning

- $D_m$ is the jump set related to measurement jumps;
- $D_{01}$ is the jump set related to measurement-to-movement mode switches;
- $D_{10}$ is the jump set related to movement-to-measurement mode switches, or *reset* jumps;
- $C_0$ is the flow set for measurement mode;
- $C_1$ is the flow set for movement mode.

The jump and flow set of the system is then obtained by unions of these sets:

$$C := C_0 \cup C_1 \subset \mathcal{X} \qquad D := D_m \cup D_{01} \cup D_{10} \tag{2.76}$$

We state that the above sets $D_m$, $C_0$, $D_{01}$, $C_1$, $D_{10}$ are closed, and a proof of this fact is given in Proposition A.3. Consequently, $C$ and $D$, being the union of closed sets, are closed.

The jump maps are defined as

$$g_m(x,\nu) = \begin{bmatrix} 0 & d & M_E(\mu, E, d+\nu) & N_E(E) & 0 \end{bmatrix}^T \qquad x \in D_m \tag{2.77}$$

$$g_{01}(x,\nu) = \begin{bmatrix} 0 & d & \mu & E & 1 \end{bmatrix}^T \qquad x \in D_{01} \tag{2.78}$$

$$g_{10}(x,\nu) = \begin{bmatrix} 0 & d & \mu & 1 & 0 \end{bmatrix}^T \qquad x \in D_{10} \tag{2.79}$$

where $\nu \sim \mathcal{N}(0, \sigma_d^2)$ is sampled at each jump and represents the i.i.d. Gaussian measurement errors, and the functions $M_E$ and $N_E$ are defined to compact the notation and represent the update expressions of the RBE.

Their derivation starts from defining the functions $M(\mu, P, \tilde{d})$ and $N(P)$, which serve the same purpose with the non-encoded representation of the variance $P$, and originate from Equation (2.29):

$$M(\mu, P, \tilde{d}) := \frac{\sigma_d^2}{P+\sigma_d^2}\mu + \frac{P}{P+\sigma_d^2}\tilde{d} \qquad = \mu^+ \tag{2.80}$$

$$N(P) := \frac{\sigma_d^2}{P+\sigma_d^2}P \qquad = P^+ \tag{2.81}$$

then, applying the change of variables of Equation (2.69) from $P$ to $E$ for the arguments, from $E$ to $P$ for the domain of $N(P)$, and taking care of the needed extensions for continuity, we obtain

$$M_E(\mu, E, \tilde{d}) := \begin{cases} \frac{-\sigma_d^2 \ln(E)}{1-\sigma_d^2 \ln(E)}\mu + \frac{1}{1-\sigma_d^2 \ln(E)}\tilde{d} & \text{if } E > 0 \\ \mu & \text{if } E = 0 \end{cases} \tag{2.82}$$

$$N_E(E) := e^{-1/\sigma_d^2} E \tag{2.83}$$

which are used above in Equation (2.77) for defining the jump maps.

We finally define the flow vector field $f(x)$ as follows

$$f(x) = \begin{bmatrix} 1-q \\ -\alpha\mu q \\ -\alpha\mu q \\ \varepsilon q \\ 0 \end{bmatrix} \in \mathbb{R}^5 \qquad x \in C_0 \cup C_1 \tag{2.84}$$

With reference to the definition of the state vector $x$ in Equation (2.68), observe that

- $f_\tau(x) = 1 - q$ makes the timer increase only during measurement mode, where $q = 0$;
- $f_d(x)$ and $f_\mu(x)$ are both equal to $-u(t)q$, so the movement "activates", impacting on the agent-target distance $d$ and updating the estimator state $\mu$, only in movement mode, that is when $q = 1$;
- $f_q(x) \equiv 0$ as the current mode only changes during jumps.

Moreover, notice that $f_E(x) = \varepsilon q$. This is necessary to avoid the system getting stuck in the following subset of the state space:

$$\mathcal{F} = \{x \in C_1 \mid \tau = 0, E = 0, q = 1\} \tag{2.85}$$

$$= \{0\} \times \mathbb{R} \times \mathbb{R} \times \{0\} \times \{1\} \tag{2.86}$$

$$\subset C_1 \tag{2.87}$$

In fact, without the above correction, the above subset would be forward-invariant: the condition for the system to exit $C_1$ is $x(t_s) \in D_{10}$ for some $t_s$. However, considering the definitions of the set in (2.75), we have

$$x \in D_{10} \implies E \geq e^{-b^2/\mu^2} > 0 \quad \forall b > 0, \mu \in \mathbb{R} \tag{2.88}$$

so we can write $x \in D_{10} \implies E > 0$, the logical inverse of which is[4]

$$E = 0 \implies x \notin D_{10} \tag{2.89}$$

and then it is clear that since

$$x \in \mathcal{F} \implies E = 0 \implies x \notin D_{10} \tag{2.90}$$

the system will never jump outside of $\mathcal{F} \subset C_1$. The term $f_E(x) = \varepsilon q$ breaks the forward-invariance of $\mathcal{F}$ for any choice of $\varepsilon > 0$. We can then choose an arbitrarily small value of $\varepsilon$ and argue that this additional dynamical component does not impact on the performance of the system (more specifically, on the behavior of the estimator).

It is only left to formally define the distribution function

$$\lambda : \mathbf{B}(\mathbb{R}) \to [0,1] \tag{2.91}$$

$$\omega \mapsto \lambda(\omega) = \int_{\omega} \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\frac{x}{\sigma_d^2}\right) \mathrm{d}x \tag{2.92}$$

which is a Normal distribution of mean 0 and variance $\sigma_d^2$, representing the measurement errors that are sampled at each jump.

To sum up, in equation (2.68) we define the state space of the system, equations (2.73), (2.74) and (2.75) define the flow and jump sets, in (2.77) the jump maps are defined, in (2.84) the flow map is provided and (2.91) defines the random distributions related to the jumps. This completes the description of the SHS in the form considered in [7], with the specific characteristics of not having a set-valued flow map, the distribution function not dependent on the state, and its transitions being entirely state driven (there are no spontaneous ones).

Finally, tables 2.1, 2.2 and 2.3 summarise and describe all the state variables, system parameters and controller parameters that characterize the system.

| **State variable** | **Description** | **Unit** |
|---|---|---|
| $\tau$ | Measurement timer, only enabled while $q = 0$ | s |
| $d$ | True agent-target distance | m |
| $\mu$ | Mean of the Gaussian estimate of the agent-target distance | m |
| $P$ | Variance of the Gaussian estimate of the agent-target distance | m$^2$ |
| $q$ | Operating mode: $q = 0$ measurement, $q = 1$ movement | - |

**Table 2.1:** State variables of the closed-loop SHS for the 1D task.

[4] Together with $E \geq 0$ which follows from the definition of the state variables in Equation (2.68).

| Parameter | Description | Unit | Properties |
|---|---|---|---|
| $T_m$ | Time between measurements | s | $> 0$ |
| $\sigma_d^2$ | Variance of the distance measurements | $\mathrm{m}^2$ | $> 0$ |
| $\varepsilon$ | Positive derivative of $E$ during movement | | $> 0, \ll 1$ |

**Table 2.2:** Parameters of the 1D system.

| Parameter | Description | Unit | Properties |
|---|---|---|---|
| $\alpha$ | Approach rate in movement mode | | $> 0$ |
| $b$ | Movement to measurement mode switch threshold | $-$ | $> 0$ |
| $a$ | Measurement to movement mode switch threshold | $-$ | $> b$ |

**Table 2.3:** Parameters of the 1D control law.

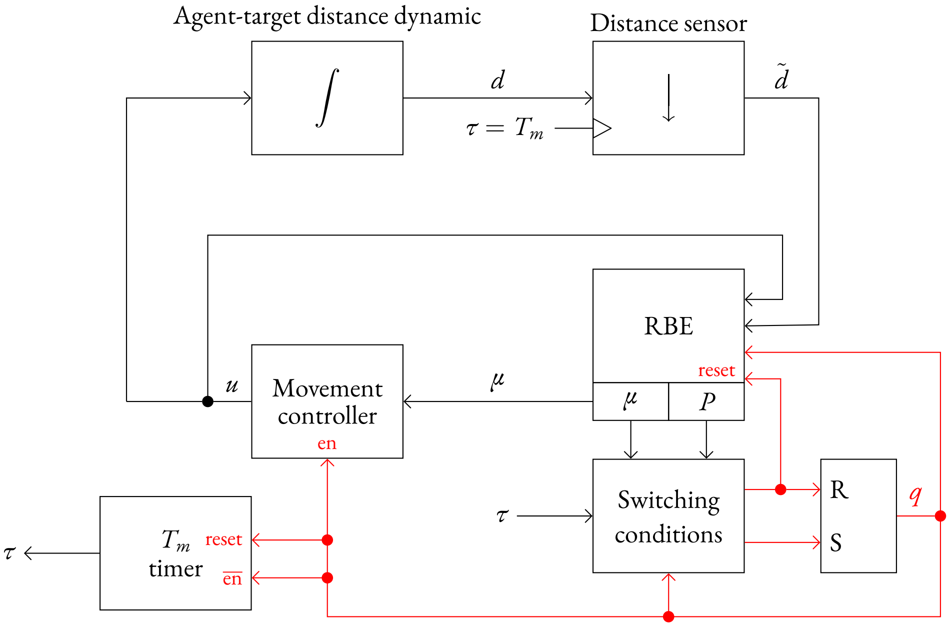


**Figure 2.3:** Block diagram of the 1D closed-loop SHS. The red color denotes logical signals. The agent-target dynamic is modeled as an integrator on the velocity control input $u$ that is provided by the control law. Then the distance sensor is modeled as a sampler block with additive noise, triggered by the timer $\tau$ reaching its maximum value $T_m$. The RBE block is at the entry of the control scheme, and is a hybrid block: measurements trigger an instantaneous state update, while during movement, the update of its state is continuous. In general, it monitors the measurements and the velocity signal, together with the current operating mode. The switching is realized with the *Switching conditions* block, that monitors $\mu$, $P$ from the estimator, and a SR flip-flop, that keeps memory of the current mode. When $q = 0$ the enable signal on the movement controller forces its output to zero, and the $T_m$ timer is forced reset (with the counter $\tau$ at 0). Otherwise, the timer ticks from 0 to $T_m$, resetting to zero by itself.

## 2.2 Simulations and behavior

While there are algorithms for generating solutions of general Stochastic Hybrid Systems [6], the proposed SHS has been simulated with an ad hoc algorithm, which samples solutions from their measurable set $\Omega$ provided an initial condition $x_0$ [7]. For a description of the algorithm, refer to Algorithm A.1 in the appendix. A faster version of the algorithm, which neglects the checks regarding $x \in C \cup D$, has also been used and is described by Algorithm A.3 in the appendix. The implementation shown in listing A.1 covers both variants.

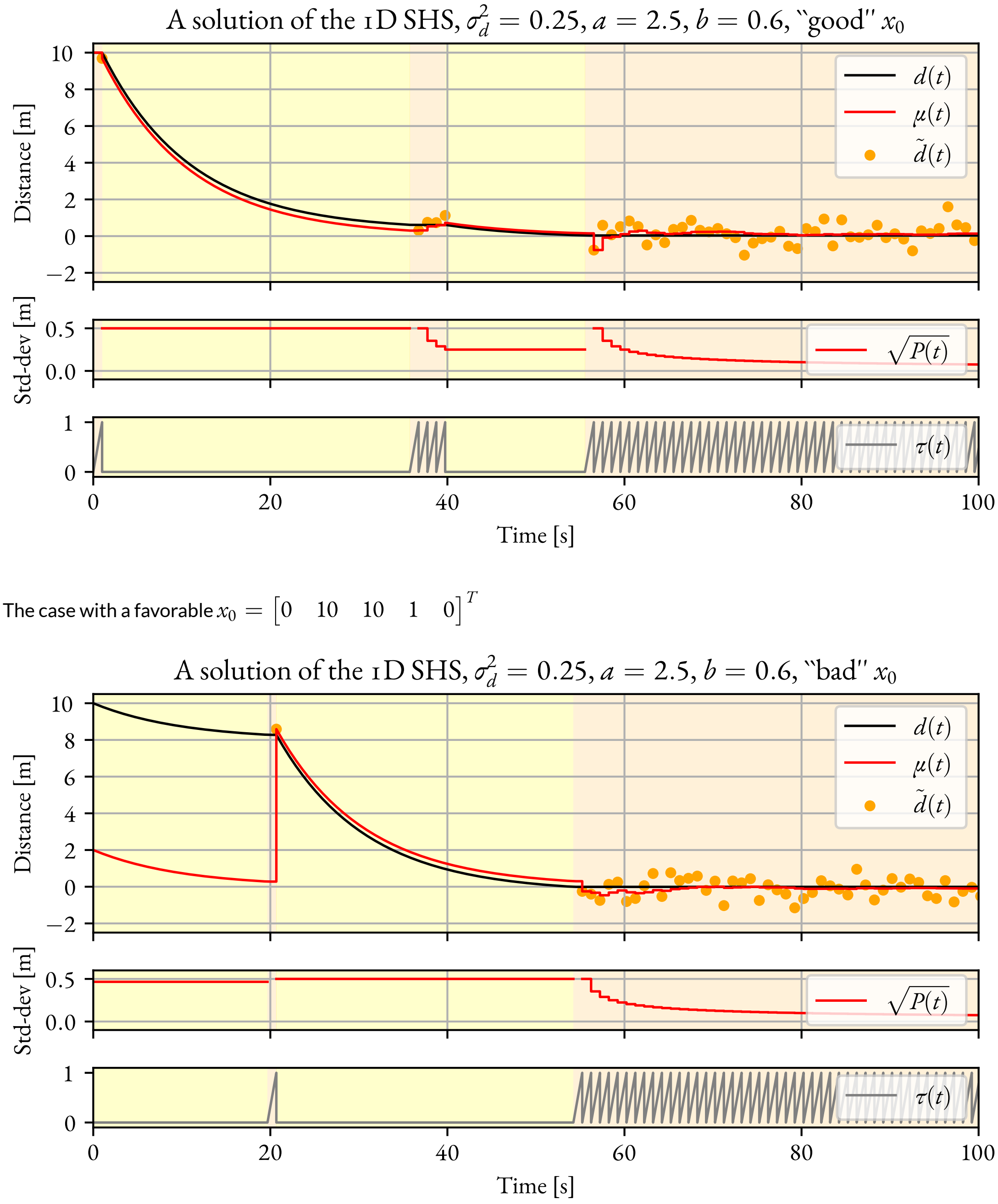
### 2.2.1 Behavior in time

The first simulation results that we present are in the shape of plots of some random solutions of the SHS in time.

Figure 2.4a shows one such experiment, with an initial choice of the most critical parameters as $a = 2.5$, $b = 0.6$, $\sigma_d^2 = 0.25$ and $\alpha = 0.1$. Please note that, unless otherwise stated, we always assume unitary $T_m = 1$s and a small $\varepsilon = 1 \times 10^{-9}$. This experiment is a good example of the typical system behavior: let us comment on the events that happen as time progresses.

- The system starts at $t_0 = 0$ in measurement mode, with no knowledge of the target position ($E(t_0) = 1$, $P(t_0) \to \infty$ which in the plot is represented by the missing segment of the $\sqrt{P(t)}$ line) and initial estimate compatible with the true distance[5]. The timer is at $\tau(t_0) = 0$ and starts increasing, awaiting the time required for a measurement to be available to elapse.
- At time $t_{s_1} = 1$ the system performs the reset measurement with the reset jump due to $x \in D_{10}$, ending up with $\mu(t_{s_1}) \simeq d(t_0)$. The timer is reset to zero, and $E^+ < 1$ (in particular $P^+ = \sigma_d^2$).
- Immediately after, the condition $x \in D_{01}$ for the measurement-to-movement mode switch is fulfilled. This happens because the mode switching conditions take into consideration the current (relatively) big distance of the system from the target, leading to a quite conservative threshold on $E$, which is already fulfilled.
- From time $t_{s_1} = 1$ to time $t_{s_2} = 35.75$, the agent moves towards the target, until the next mode switching condition $x \in D_{10}$ is met. This movement phase ends with the agent closer to the target: $d(t_{s_2}) \simeq 0.6$.
- A movement-to-measurement mode switch, or *reset* jump, happens again at $t_{s_2}$. The estimator is signaled again for the need of a reset measurement with $E^+ = 1$. The system enters the measurement mode with the timer $\tau(t_{s_2}) = 0$, which starts to increase again.
- The system stays in measurement mode from time $t_{s_2}$ to time $t_{s_3} = 39.75$, performing information fusion of the incoming measurements. Observe that $P(t)$ decreases monotonically at each measurement jump. The timer ensures the correct timing between the measurements.
- At time $t_{s_3}$ the movement mode is entered again. This time, thanks to the more accurate estimate that was obtained in the previous measurement phase, the mode-switching conditions allow the system to move closer to the target, ending up at $d(t_{s_4}) \simeq 0.05$, $t_{s_4} = 55.54$.
- At time $t_{s_4}$ the measurement mode is entered again. This time, due to the smaller distance between the agent and the target, the mode-switching conditions require a higher precision estimate before entering the next movement phase. In the limit of the 100s of total simulation time that is depicted, this does not happen, even though the variance of the estimate is decreasing at every measurement jump. The agent, however, is very close to the target at the end of the considered time frame.

[5] The initial value $\mu(t_0)$ is actually not important, since the system as such is ready for a reset measurement, as designed in subsection 2.1.6



**(a)** The case with a favorable $x_0 = \begin{bmatrix}0 & 10 & 10 & 1 & 0\end{bmatrix}^T$

**(b)** The case with a misleading $x_0 = \begin{bmatrix}0 & 10 & 2 & 0.01 & 0\end{bmatrix}^T$

**Figure 2.4:** Behavior in time of the 1D system, obtained by sampling a random solution. All state variables are plotted, with the current mode $q(t)$ being depicted through the differently colored background segments. The parameters are chosen as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}, \sigma_d^2 = 0.25$, $a$ and $b$ as in the plot title. The ``good'' and ``bad'' $x_0$ cases, with reference to the initial condition, are compared.

We can then observe that the dynamic we designed is able to approach the target gradually, switching between the two modes and trying (depending on the parameters choices) to react early when enough information is available. The true distance is improved upon, cycle after cycle. At the beginning, big improvements are achieved, while later progressively smaller adjustments are performed.

Figure 2.4b shows instead the results of an experiment where a "bad" initial condition was used. In particular, the initial estimate has been initialized quite differently with respect to the true agent-target distance. This scenario is the one previously described in subsection 2.1.6 that motivated the introduction of the *reset* mechanism. Notice in the plot that while the agent initially ends up in a bad location (at the end of the first movement phase), it is able to quickly recover and approach the target arguably well, in a reasonable time. Please note that the two experiments of Figure 2.4 depict only two random solutions of the closed-loop stochastic hybrid system, so the fact that the distance ends up closer after only one movement phase following the reset, with respect to the two movement phases of the previously discussed experiment, is just a random outcome and is not indicative of better performance with bad initial conditions in general.

In fact, it is important, when observing some quantitative aspect of the system behavior, to consider the expected value over all possible solutions. In the next section, we will indeed try to understand what happens to the localization performance with respect to the choices of the parameters $a$, $b$ for various choices of $\sigma_d^2$.

### 2.2.2 Effect of parameters on task completion metrics

Let us assume that $\sigma_d^2$, which characterizes the quality of the measurements, is given and fixed. As stated before, the time required for a measurement to be ready is $T_m = 1\text{s}$ and $\alpha = 0.1$, $\varepsilon = 1 \times 10^{-9}$.

It is interesting to study the effect of the controller parameters $a$ and $b$ on some performance metrics of choice. For this purpose, two metrics will be considered:

1. the final distance, after some fixed time, between the agent and the target;
2. the time required for the agent to switch back to measurement mode with the true distance below a given threshold.

The motivation for experimentally measuring this last quantity originates from the possibility of the agent, tasked to also rescue the target, to be provided with some kind of sensing unit capable of detecting with very high confidence the presence of the target in a given short range[6]. In such case we might consider the rescue task to be completed.

For the first metric, we fix the final time $t_f = 100\text{s}$ and estimate the expected value of the final distance as follows:

$$\mathbb{E}\left[|d(t_f)|\right] = \mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{S}_{\mathbf{r}}(x_0)}\left[\sup_j \left\{|\mathbf{x}_d(t_f, j)|\right\}\right] \tag{2.93}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{S}_{\mathbf{r}}(x_0)}\left[|\mathbf{x}_d(t_f)|\right] \tag{2.94}$$

$$\simeq \frac{1}{N}\sum_{i=1}^{N}|\mathbf{x}_d(\omega_i)(t_f)| \tag{2.95}$$

[6] This could be visual and/or infrared thermal sensing, for instance.

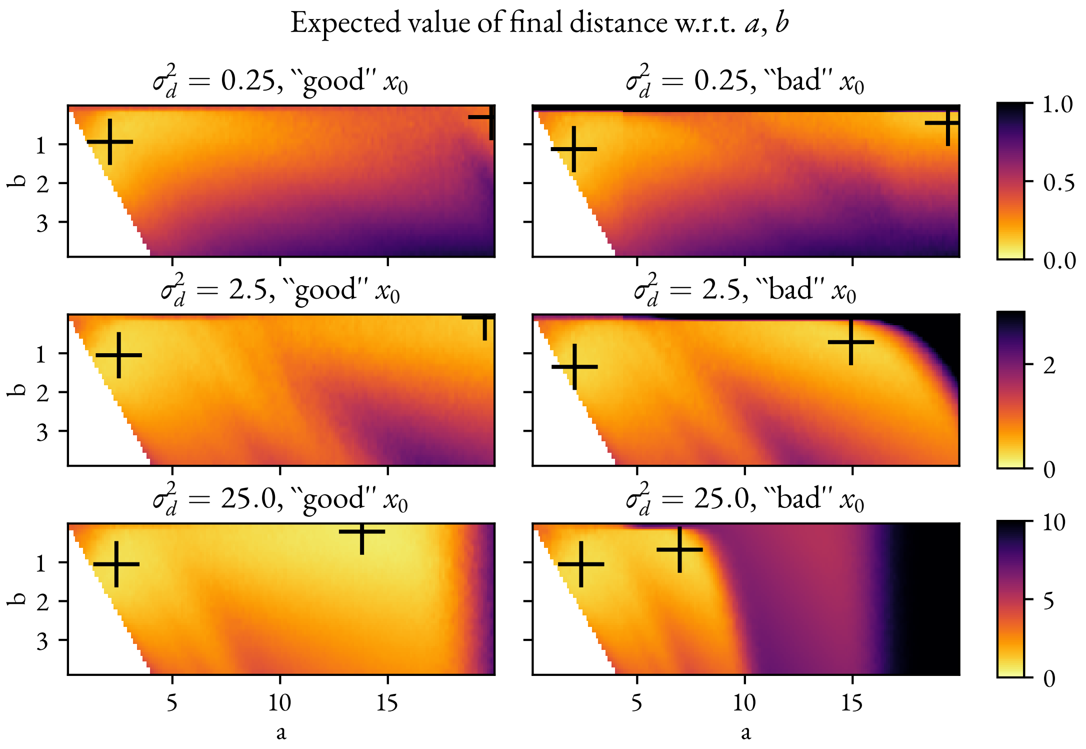


**Figure 2.5:** Expected value of the 1D final distance $\mathbb{E}\left[|d(t_f)|\right]$, with respect to the choice of the two parameters $a, b$, in the presence of low, medium and high measurement noise $\sigma_d^2$. The final time has been chosen as $t_f = 100$s and the parameters as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}, T_m = 1$s. The "good" and "bad" $x_0$ cases, with reference to the initial condition, are compared, where the good $x_0 = \begin{bmatrix} 0 & 10 & 10 & 1 & 0 \end{bmatrix}^T$ and the bad $x_0 = \begin{bmatrix} 0 & 10 & 2 & 0.01 & 0 \end{bmatrix}^T$. The crosses denote the location of the local minima in the considered parameter range.

where $\mathbf{x} \in \mathcal{S}_\mathbf{r}(x_0)$ is a mapping from $\Omega$ to the set of hybrid arcs and represents a random solution starting from $x_0$ [7]. In (2.94) we omitted the sup operation and the indexing on $j$ since, in the system we are generating solutions of, there are no jumps that change the value of the distance: $x(t) \in D \implies x_d(t^+) = x_d(t)$. The samples $\omega_i \in \Omega$ in (2.95) are obtained by randomly sampling values from the distribution function $\lambda(\cdot)$ defined in Equation (2.91) at each jump, as performed by Algorithm A.3 which is used for these Monte Carlo estimations. In all experiments, it was chosen $N = 500$.

Figure 2.5 plots the resulting value of the estimate described by Equation (2.95) for different values of the parameters $a$ and $b$ in a limited range around the origin, in the three cases of low, medium and high measurement noise. A cross denotes the location where the minimum value has been measured.

As for the second metric, the procedure is similar. This time, however, the Monte Carlo method approximates

$$\mathop{\mathbb{E}}_{\mathbf{x}\in\mathcal{S}_\mathbf{r}(x_0)}\left[t_R := \min\left\{t \geq 0 \,\middle|\, \mathbf{x}_q(t) = 0, |\mathbf{x}_d(t)| \leq d_R\right\}\right] \tag{2.96}$$

that is, the "reach time" $t_R$ defined as the first time instant (from the task starting time) where the agent is within

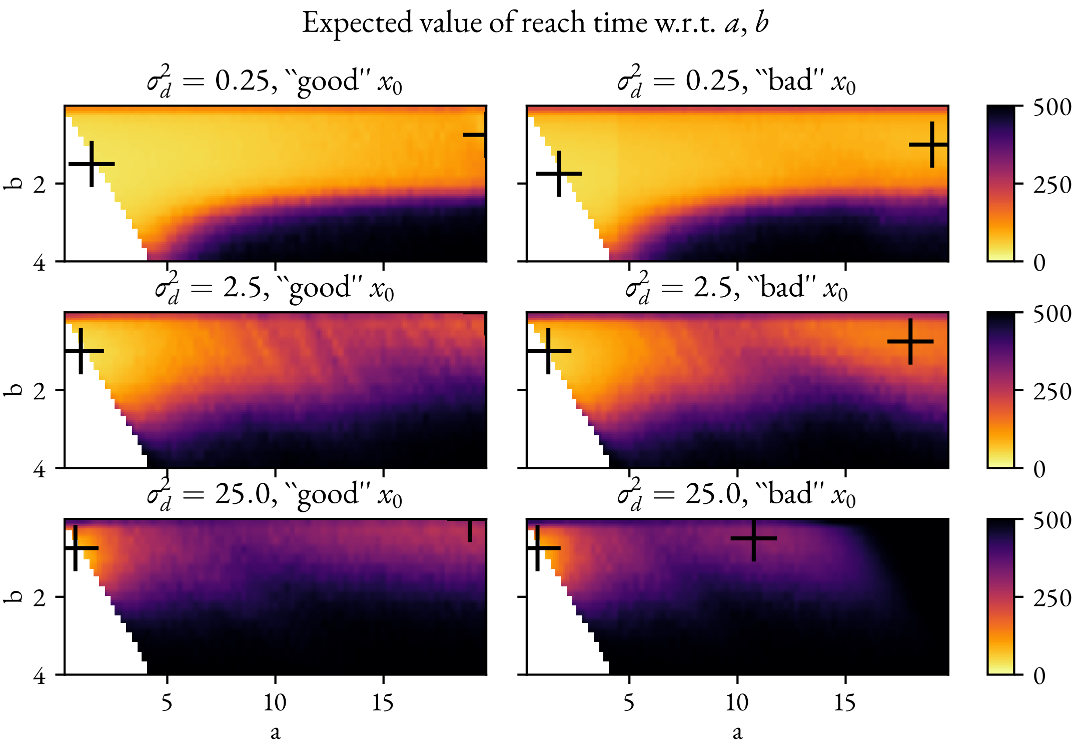


**Figure 2.6:** Expected value of the 1D approach time $\mathbb{E}[t_R], t_R := \min\{t \geq 0 \,|\, q(t) = 0, |d(t)| \leq d_R\}$, where $d_R = 0.5\text{m}$, with respect to the choice of the two parameters $a, b$, in the presence of low, medium and high measurement noise $\sigma_d^2$. The maximum operating time is bounded at $500\text{s}$ and the parameters chosen as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}$ and $T_m = 1\text{s}$. The "good" and "bad" $x_0$ cases, with reference to the initial condition, are compared, where the good $x_0 = \begin{bmatrix} 0 & 10 & 10 & 1 & 0 \end{bmatrix}^T$ and the bad $x_0 = \begin{bmatrix} 0 & 10 & 2 & 0.01 & 0 \end{bmatrix}^T$. The crosses denote the location of the local minima in the considered parameter range.

the chosen range $d_R$ of the target, and the measurement mode is active. In this way, we are observing how much time is needed for the rescue task to complete, given a fixed final distance threshold. For all experiments, it has been chosen $d_R = 0.5\text{m}$.

Figure 2.6 plots the resulting value of the estimate of Equation (2.96) for different values of the parameters $a$ and $b$, in the three cases of low, medium and high measurement noise. As before, a cross denotes the location where the minimum value has been measured.

Finally, note that, for each scenario and for both metrics, two initial conditions are considered: a favorable one with a meaningful estimator initial state, and a misleading one with a bad estimator state.

### 2.2.3 Observations

We now comment on the experimental results of the previous section.

Looking at the expected value of the two performance criteria in figures 2.5 and 2.6, we visually observe that there are two local minima in the considered range of the parameter space $(a, b)$[7].

The leftmost local minimum location does not look to be dependent on the kind of initial condition, but its position does instead change based on the value of $\sigma_d^2$. The rightmost one, instead, changes its location also based on whether $x_0$ is favorable or not. Figure 2.7 plots a solution of the system with $\sigma_d^2 = 25$ and this second set of locally optimal parameters, obtained in the final distance experiment of Figure 2.5 with $\sigma_d^2 = 25$ and a good initial condition.

We observe indeed that this set of parameters leads to a good performing execution with a favorable $x_0$ (Figure 2.7a), but does not perform as well with a misleading initial condition (Figure 2.7b). This can be intuitively explained by the fact that such a choice, in particular with $a \gg 0$ and $b \ll a$, tune the system so that the measurement phase lasts longer[8] and the movement phase tries to stop the agent very close to the estimated position of the target. It should be clear that such behavior is well suited if the system is correctly initialized, with the variance $P(0)$ encoding the real variance of the estimate $\mu(0)$ (or, as in the experiment of Figure 2.7a, with $x(0)$ being a reset state). If the initial estimator state is instead not representing the true mean and variance, like previously discussed in subsection 2.1.6, the system will need to perform a reset jump before being able to correctly perform the task as designed. A big value of the parameter $a$ increases the time required for this to happen, and leads to worser performance (in reach time and final distance at a given limit time) as we can clearly observe in Figure 2.7b.

The leftmost local optimum, instead, tunes the control law differently, leading to a typical execution which solves the issue that was just described. Now that $a$ is not much bigger than $b$, the mode switching happens more often. Figure 2.8 plots two solutions of the closed-loop SHS with $\sigma_d^2 = 25$ and the locally optimal parameters $a = 2.4$ and $b = 1.1$ from the experiments with $\sigma_d^2 = 25$ of Figure 2.5.

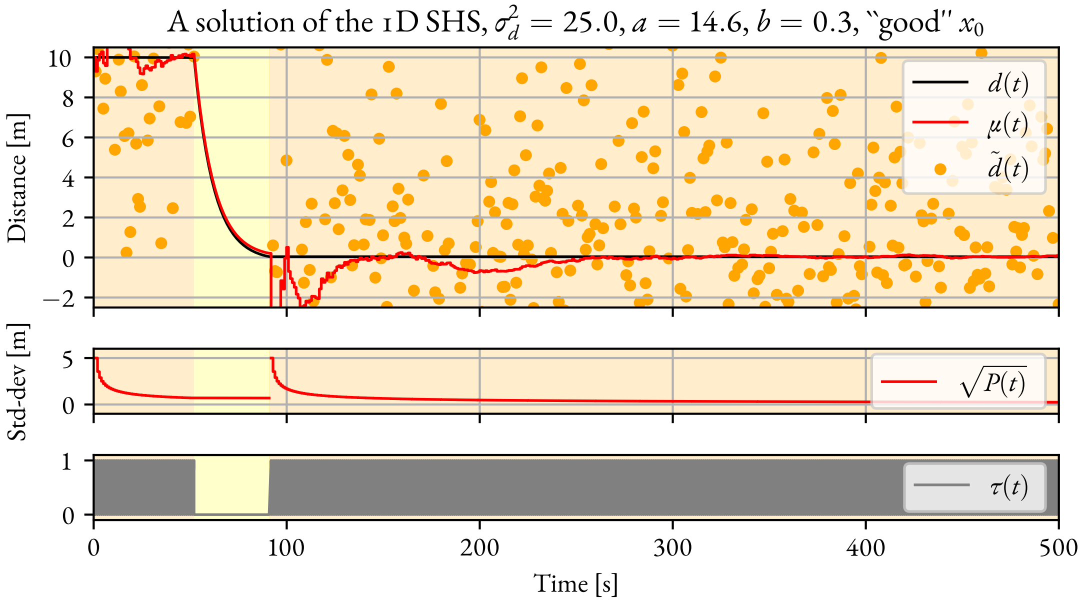
We observe that this leads to more frequent reset jumps, allowing the system to recover earlier from a misleading $x_0$. This can be observed in Figure 2.8b where within the initial 15s, a reset happens which leads to a (relative to the previous case) quickly converging distance dynamic. We also observe that, by comparing the two random solutions of Figure 2.7a and Figure 2.8a, the performance of the task execution is qualitatively slightly worse, in this specific instance, with this second set of parameters. We may argue that this is not a coincidence, since it makes sense that this local minimum for the two classes of initial conditions may only be the global minimum for the misleading $x_0$ case.

Another observation is that, in the *reach time* experiment, the leftmost local optimum is always very close to the diagonal $a = b$ (of course it must be $a > b$). Intuitively a small difference $\delta := a - b \ll 1$ leads to more frequent switching, and the motivation for the system to prefer such behavior is that, given the experiment setup previously explained in section 2.2, this leads to statistically more opportunities for the task to be considered complete: it is in fact required that the system switches back to measurement mode for this condition to happen. In Figure 2.9 we observe this behavior.

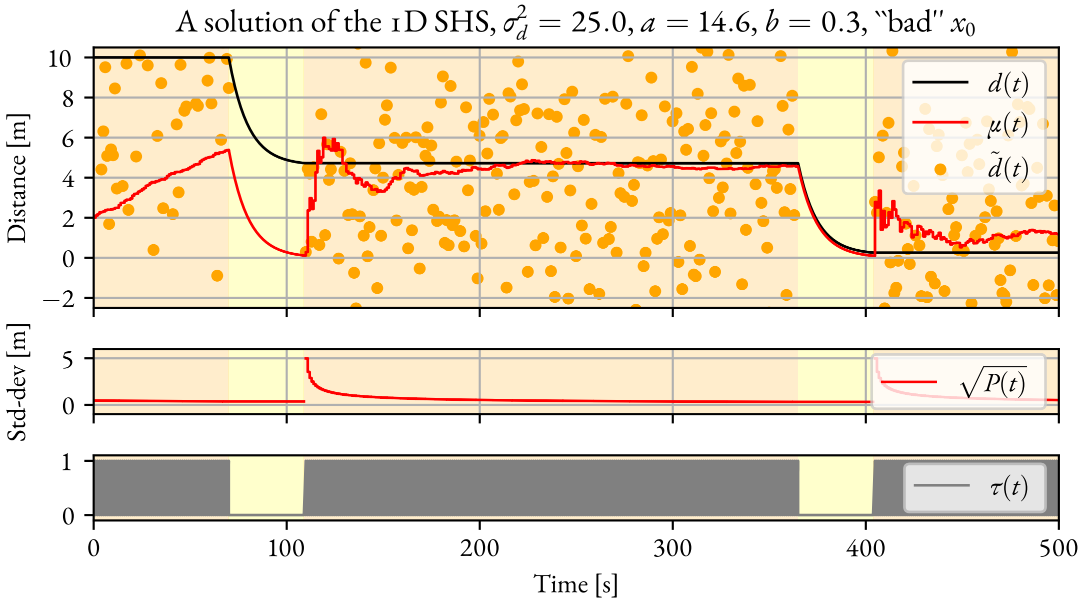
Figure 2.9b in particular depicts an interesting way of acting, where the time spent in measurement mode is very short, and as such, the movement mode starts with a very high variance of the estimated distance. We may describe the approach that arises with such set of parameters as one resembling a random walk around the target $d = 0$, with frequent switching to maximize the chance of terminating the task. Moreover, looking at the location

[7] This is especially well visible in the final distance experiment, and for high values of $\sigma_d^2$ in the *reach time* experiment.

[8] More specifically, so that the variance of the estimate grows smaller, thus requiring a more accurate estimated value.

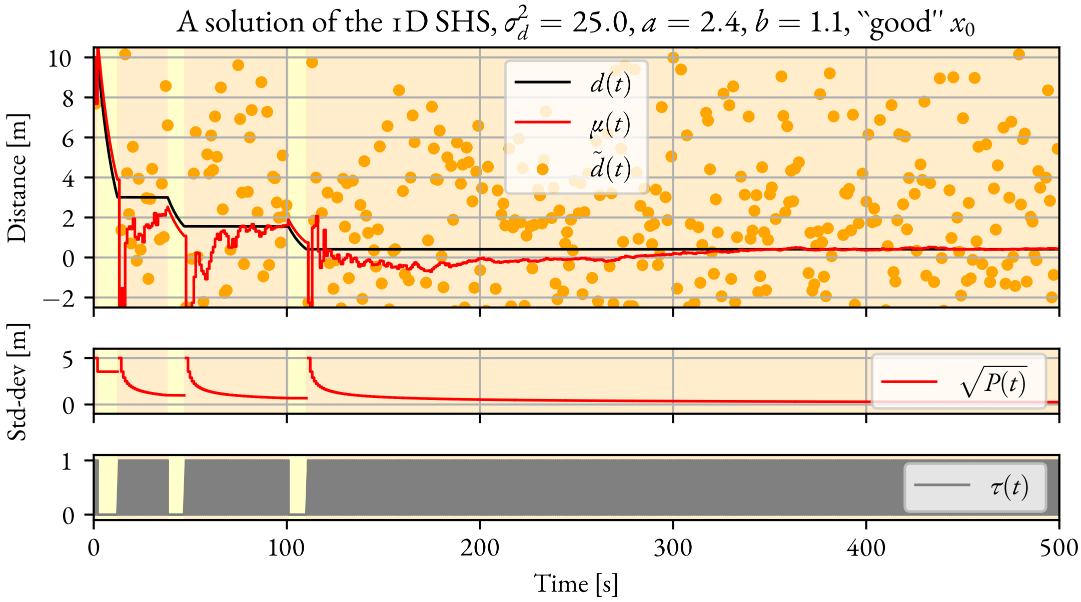


**(a)** The case with a favorable $x_0 = \begin{bmatrix} 0 & 10 & 10 & 1 & 0 \end{bmatrix}^T$

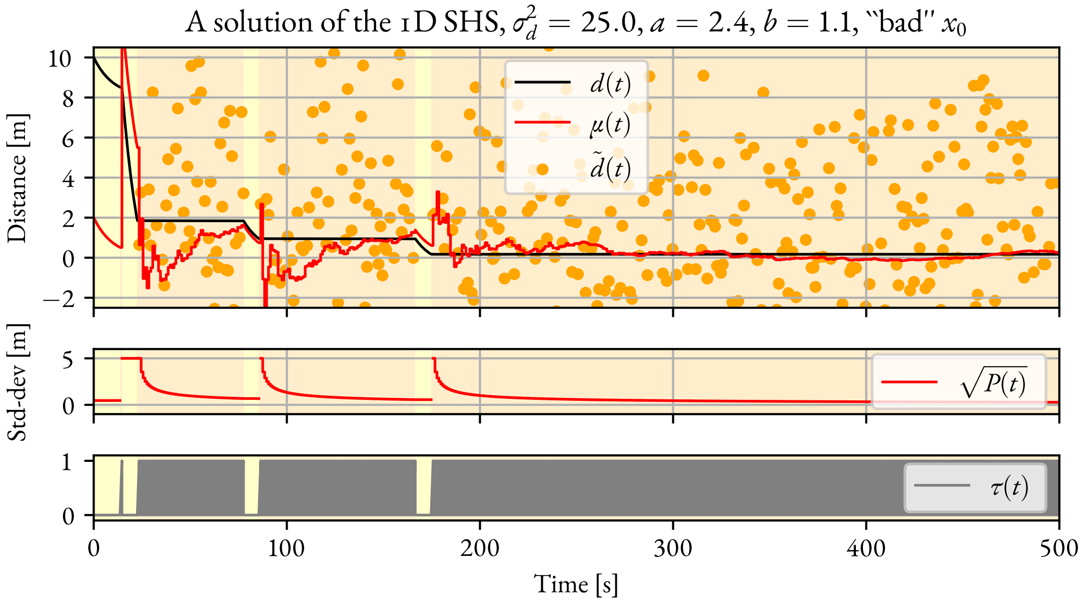


**(b)** The case with a misleading $x_0 = \begin{bmatrix} 0 & 10 & 2 & 0.01 & 0 \end{bmatrix}^T$

**Figure 2.7:** Behavior in time of the 1D system with high measurement noise and the first optimal set of parameters $a, b$ obtained by the experiment of Figure 2.5 in the case $\sigma_d^2 = 25$. The parameters are chosen as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}$, $\sigma_d^2 = 0.25$, $a$ and $b$ as in the plot title. The ``good'' and ``bad'' $x_0$ cases are compared, and by looking at these two random solution samples we observe that this set of parameters is only optimal for the case of favorable $x_0$.

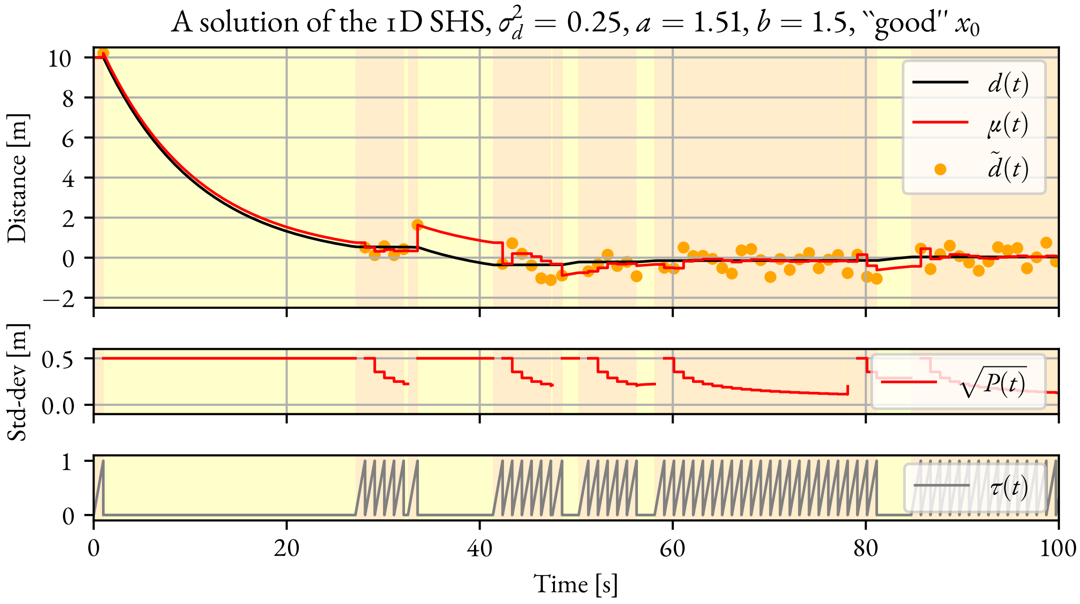


**(a)** The case with a favorable $x_0 = \begin{bmatrix} 0 & 10 & 10 & 1 & 0 \end{bmatrix}^T$

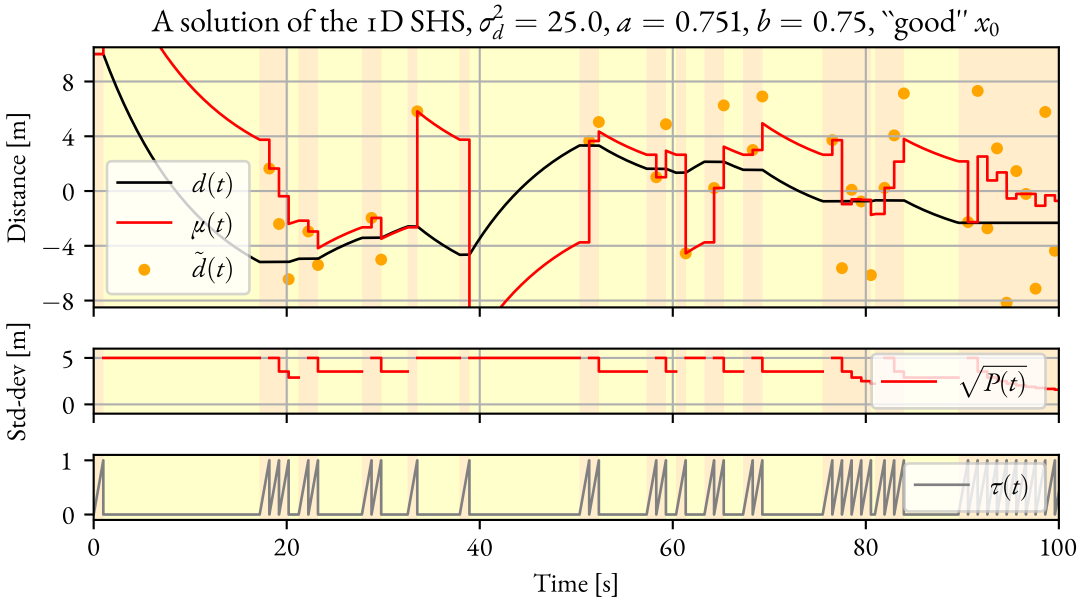


**(b)** The case with a misleading $x_0 = \begin{bmatrix} 0 & 10 & 2 & 0.01 & 0 \end{bmatrix}^T$

**Figure 2.8:** Behavior in time of the 1D system with high measurement noise and the first optimal set of parameters $a, b$ obtained by the experiment of Figure 2.5 in the case $\sigma_d^2 = 25$. The parameters are chosen as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}$, $\sigma_d^2 = 0.25$, $a$ and $b$ as in the plot title. The ``good'' and ``bad'' $x_0$ cases are compared, and by looking at these two random solution samples we observe that this set of parameters makes the system behave well with both classes of initial conditions.



**(a)** The case with low measurement noise.



**(b)** The case with high measurement noise

**Figure 2.9:** Behavior in time of the 1D system with low and high measurement noise with the optimal sets of parameters $a, b$ for minimal reach time, obtained by the experiments of Figure 2.6 in the cases $\sigma_d^2 = 0.25$ and $\sigma_d^2 = 25$.

of the leftmost local minima in Figure 2.6 across the different values of $\sigma_d^2$, we notice that the optimal values of $a \sim b$ decrease as $\sigma_d^2$ increases. This might initially seem counterintuitive, since we would expect the system to require a higher estimate accuracy ($a \gg 1$) with an increasing measurement noise variance, but can now be explained by the appearance of this second style of operation that we observe in Figure 2.9b, which is measurably better at minimizing the reach time as defined in Equation (2.96).

## 2.3 Stochastic Hybrid Basic Conditions

We proceed by proving that the Hybrid Basic Conditions [5] [7] hold for the system described in section 2.1.

**Proposition 1** *For the system with data (2.68), (2.76), (2.77), (2.84) and (2.91) the Hybrid Basic Conditions hold, namely*

1. *The sets $C$ and $D$ are closed.*
2. *The set-valued mapping $F: \mathbb{R}^5 \rightrightarrows \mathbb{R}^5$ is outer semicontinuous, locally bounded, and for each $x \in C$ the value $F(x)$ is nonempty and convex.*
3. *The set-valued mapping $G: \mathbb{R}^5 \times \mathbb{R} \rightrightarrows \mathbb{R}^5$ is locally bounded and $x \mapsto G(x, v)$ is outer semicontinuous $\forall v \in \mathbb{R}$.*

Proof We proceed by proving the three points

1. Proven by Proposition A.3.
2. We construct the set-valued mapping $F(x) := \{f(x)\}\ \forall x \in C$ and observe that $\forall x \in C$ the value $F(x)$ is trivially nonempty and convex. Moreover, being $F(x)$ obtained from the continuous, locally bounded function

$$f(x) = \begin{bmatrix}1-q & -\alpha\mu q & -\alpha\mu q & \varepsilon q & 0\end{bmatrix}^T \in \mathbb{R}^5 \qquad x \in C = C_0 \cup C_1 \tag{2.97}$$

   it is outer semicontinuous and locally bounded [5, section 5.2].
3. Recall the definitions of the jump sets in (2.76):

$$D_m = \{T_m\} \times \mathbb{R} \times \{E \geq e^{-a^2/\mu^2}\} \times \{0\} \tag{2.98}$$
$$D_{01} = \{0\} \times \mathbb{R} \times \{E \leq e^{-a^2/\mu^2}\} \times \{0\} \tag{2.99}$$
$$D_{10} = \{0\} \times \mathbb{R} \times \{E \geq e^{-b^2/\mu^2}\} \times \{1\} \tag{2.100}$$

   and notice that they are pair-wise disjoint. We then define the set-valued jump map $G(x, v)$ as follows

$$G(x, v) = \begin{cases} \{g_m(x, v)\} & \text{if } x \in D_m \\ \{g_{01}(x, v)\} & \text{if } x \in D_{01} \\ \{g_{10}(x, v)\} & \text{if } x \in D_{10} \\ \emptyset & \text{otherwise} \end{cases} \tag{2.101}$$

with the definitions of the jump maps previously given in (2.77). It is then clear that the value of $G(x, \nu)$ is either empty or singleton $\forall x, \nu \in \mathrm{dom}(G)$. Consider

$$g_{01}(x, \nu) = \begin{bmatrix} 0 & d & \mu & E & 1 \end{bmatrix}^T \qquad g_{10}(x, \nu) = \begin{bmatrix} 0 & d & \mu & 1 & 0 \end{bmatrix}^T \tag{2.102}$$

and notice they are continuous and locally bounded in $x$, $\nu$. Then, consider

$$g_m(x, \nu) = \begin{bmatrix} 0 & d & M_E(\mu, E, d + \nu) & N_E(E) & 0 \end{bmatrix}^T \qquad x \in D_m \tag{2.103}$$

and recall the definitions

$$M_E(\mu, E, \tilde{d}) := \begin{cases} \frac{-\sigma_d^2 \ln(E)}{1-\sigma_d^2 \ln(E)}\mu + \frac{1}{1-\sigma_d^2 \ln(E)}\tilde{d} & \text{if } E > 0 \\ \mu & \text{if } E = 0 \end{cases} \tag{2.104}$$

$$N_E(E) := e^{-1/\sigma_d^2} E \tag{2.105}$$

to verify that $N_E(E)$ is continuous and locally bounded $\forall E \in [0, 1]$ (recall $\sigma_d^2 > 0$), and

$$\lim_{E \to 0^+} \frac{-\sigma_d^2 \ln(E)}{1 - \sigma_d^2 \ln(E)}\mu + \frac{1}{1 - \sigma_d^2 \ln(E)}\tilde{d} = \mu \tag{2.106}$$

We notice that the two fractions are always $\in [0, 1]$, $\forall E \in [0, 1]$, and conclude that $M_E$ is locally bounded in $x$, $\nu$. Moreover, for all given $\nu$, $x \mapsto M_E(x, \nu)$ is continuous. We conclude that $G(x, \nu)$ is a locally bounded set-valued mapping and is outer semicontinuous w.r.t. $x$, as requested. ■

Moreover, we prove that the Stochastic Hybrid Basic Condition holds [7].

**Proposition 2** *For the system with data (2.68), (2.76), (2.77), (2.84) and (2.91) the Stochastic Hybrid Basic Condition holds, namely the set-valued mapping $\nu \mapsto \mathrm{graph}(G(\cdot, \nu))$ is measurable.*

Proof We have that $G : \mathbb{R}^n \times \mathbb{R} \rightrightarrows \mathbb{R}^n$ is outer semicontinuous, then $\nu \mapsto \mathrm{graph}(G(\cdot, \nu))$ is measurable [7, appendix A.2]. ■

# 3

# Two-dimensional task

## 3.1 Extending the one-dimensional model

The work of the previous chapter has also been extended to the 2D plane, leveraging most of the results previously found. In this chapter, we explain the changes needed to obtain a closed-loop model, similar to the previous one, but in two dimensions.

### 3.1.1 Task setup in $\mathbb{R}^2$

We now have the agent and target positions as vectors, together with the angle representing the current agent bearing:

$$\vec{x}_A(t) = \begin{bmatrix} x_A(t) \\ y_A(t) \end{bmatrix} \in \mathbb{R}^2 \qquad \theta_A(t) \in \mathbb{S}^1 \tag{3.1}$$

$$\vec{x}_T(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \end{bmatrix} \in \mathbb{R}^2 \tag{3.2}$$

The agent-target distance is now defined as the vector

$$\vec{d}(t) = \vec{x}_T(t) - \vec{x}_A(t) \in \mathbb{R}^2 \tag{3.3}$$

and we denote the length of this vector, which is the scalar distance, as

$$d(t) = \|\vec{d}(t)\|_2 \tag{3.4}$$

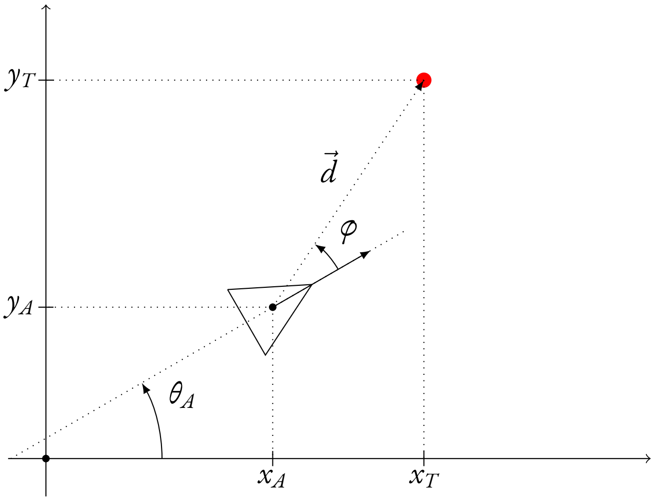


**Figure 3.1:** 2D task setup in the plane, where the agent is located at $(x_A, y_A)$ and the target at $(x_T, y_T)$. The agent bearing is denoted by $\theta_A$, and $\vec{d}$ is the agent-target distance 2D vector. $\varphi$ is the angular pointing error. The agent model is assumed to be that of the unicycle.

together with the pointing error between the agent and the target:

$$\varphi(t) = \text{atan2}\left(\vec{d}_y(t), \vec{d}_x(t)\right) - \theta(t) \tag{3.5}$$

Figure 3.1 shows all these quantities geometrically in the two-dimensional plane, which is the context of this 2D version of the localization task. We assume the sensing unit to provide the following noisy measurements:

$$\tilde{d}(t) \sim \mathcal{N}(d(t), \sigma_d^2) \tag{3.6}$$

$$\tilde{\varphi}(t) \sim \text{vonMises}(\varphi(t), \kappa_\varphi) \tag{3.7}$$

where $\kappa_\varphi$ characterizes the measurement quality of the pointing error between the agent and the target. The continuous distribution of the angle measurements is chosen as the von Mises, which is defined on $\mathbb{S}^1$, and is described in a later section. We already hint at the fact that the choice of this distribution is motivated by previous work in the literature regarding audio sensing [13], and we will later expand upon this.

### 3.1.2 Control input and movement action

The agent actuation now includes the driving velocity $v(t)$ and the steering velocity $\omega(t)$ in the agent's body frame. We are in fact assuming the agent dynamic to be represented by the unicycle model [22] and Figure 3.2 shows how the two inputs act on the agent subsystem.

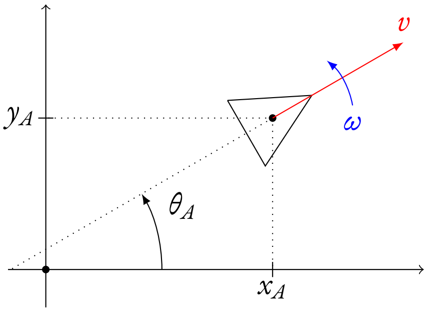


**Figure 3.2:** Actuation of the 2D agent, modeled as the unicycle. The scalar $v$ specifies the driving velocity along the agent's sagittal axis, depicted in red in the picture. The scalar $\omega$ denotes the steering velocity, that is how fast the agent bearing $\theta_A$ changes.

The dynamical equations of the agent are then

$$\begin{cases} \dot{x}_A(t) = \cos(\theta_A(t))v(t) \\ \dot{y}_A(t) = \sin(\theta_A(t))v(t) \\ \dot{\theta}_A(t) = \omega(t) \end{cases} \tag{3.8}$$

and we obtain the following dynamic in $d(t)$:

$$\begin{aligned} \dot{d} &= \frac{\mathrm{d}}{\mathrm{dt}}\left[\sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}\right] && (3.9) \\ &= \frac{1}{2}\frac{1}{d}\left(\frac{\mathrm{d}}{\mathrm{dt}}\left[(x_T - x_A)^2\right] + \frac{\mathrm{d}}{\mathrm{dt}}\left[(y_T - y_A)^2\right]\right) && (3.10) \\ &= -\frac{1}{d}\left((x_T - x_A)\dot{x}_T + (y_T - y_A)\dot{y}_T\right) && (3.11) \\ &= -\frac{1}{d}\left(d_x \cos\theta_A + d_y \sin\theta_A\right) v && (3.12) \\ &= -\left(\frac{\vec{d}}{d}\right)^T \cdot \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix} v && (3.13) \\ &= -\cos(\varphi)\, v && (3.14) \end{aligned}$$

since $\varphi(t)$ is the angle between the vector $\vec{d}(t)$ and the agent pointing direction and both vectors in the inner product of Equation (3.13) are of unit length.

The dynamic of $\varphi(t)$ is instead obtained, starting from Equation (3.5), and for $\vec{d(t)} \neq \vec{0}$, as follows:

$$\begin{aligned}
\dot{\varphi} &= \frac{\mathrm{d}}{\mathrm{dt}}\left[\operatorname{atan2}\left(d_y, d_x\right)\right] - \dot{\theta}_A && (3.15)\\
&= \frac{d_y\dot{x}_A - d_x\dot{y}_A}{d_x^2 + d_y^2} - \dot{\theta}_A && (3.16)\\
&= \frac{v(\cos(\theta_A)d_y - \sin(\theta_A)d_x)}{d^2} - \dot{\theta}_A && (3.17)\\
&= \frac{vd(\cos(\theta_A)\sin(\theta_A + \varphi) - \sin(\theta_A)\cos(\theta_A + \varphi))}{d^2} - \dot{\theta}_A && (3.18)\\
&= \frac{\sin(\varphi)}{d}v - \omega && (3.19)
\end{aligned}$$

where we used the definitions of subsection 3.1.1 and the additional relations

$$\begin{aligned}
d_x(t) &= d(t)\cos(\theta_A(t) + \varphi(t)) && (3.20)\\
d_y(t) &= d(t)\sin(\theta_A(t) + \varphi(t)) && (3.21)
\end{aligned}$$

followed by the trigonometric identity

$$\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta) \qquad (3.22)$$

### 3.1.3 von-Mises distribution for bearing and 2D RBE

The measured angles belong to $\mathbb{S}^1$ and, as such, adopting a distribution defined on this space is helpful. While we might initially choose to represent angles as scalars in $[0, 2\pi] \subset \mathbb{R}$, care would be required when using the operators of the real field on such quantities, especially when performing information fusion under the RBE paradigm. Using a distribution defined on $\mathbb{S}^1$ helps to alleviate this issue and also shows that, in the way we modeled the task, we can adopt different distributions for the information fusion part under the RBE framework. The von Mises distribution, also known as the circular normal distribution or Tikhonov distribution, can in fact be regarded as the circular analogue of the normal distribution on the line [23].

Moreover, previous work [13] has highlighted that the output of DOA estimation algorithms can be interpreted as a multi-modal p.d.f. on $\mathbb{S}^1$. While the referenced work by Bush and Xiang adopted Laplace density functions for fitting the DOA output data and obtaining the multi-modal p.d.f. through a linear combination of them, we could fit a linear combination of von Mises densities for the same purpose. If the sound source is only one, like in the scenario we are considering, the directional output of the sound sensing subsystem would then be a von Mises distribution that encodes the uncertain measurement of the angular error between the agent bearing and the body-frame target direction.

The density function of a von Mises random variable $X$ is defined as

$$p_X(\theta) = \frac{1}{2\pi I_0(\kappa)}\exp\left(\kappa\cos(\theta - \mu)\right) \qquad (3.23)$$

where $I_0(x)$ is the modified Bessel function of the first kind and order zero. The parameter $\mu$ coincides with the

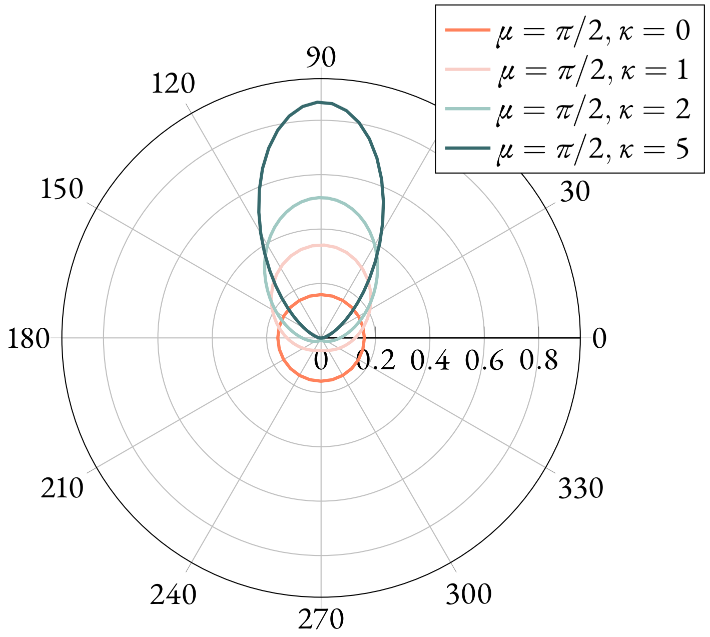


**Figure 3.3:** Examples of von Mises density functions, in polar coordinates, for different increasing values of the parameter $\kappa$, all of which are centered at $\mu = \frac{\pi}{2}$.

peak and expected value of the distribution on the circle, and the concentration parameter $\kappa$ can be assimilated to the reciprocal $\sigma^{-2}$ of the variance of a normal distribution. Figure 3.3 plots some p.d.f.s for different values of the parameters, using polar coordinates to highlight the nature of the distribution.

Notice in fact that for $\kappa = 0$ the distribution is well defined and coincides with the uniform distribution on the circle:

$$p_X(\theta)|_{\kappa=0} = \frac{1}{2\pi I_0(0)} e^0 = \frac{1}{2\pi} \tag{3.24}$$

and, for large values of $\kappa$, the density can be approximated by a Gaussian density function with $\sigma^2 = \kappa^{-1}$ [24]

$$p_X(\theta)|_{\kappa\to\infty} \sim \frac{1}{\sqrt{2\pi\kappa^{-1}}} \exp\left(-\frac{1}{2}\frac{(\theta-\mu)^2}{\kappa^{-1}}\right) \tag{3.25}$$

As stated previously, the mean of the distribution is $\mathbb{E}[X] = \mu$, while the variance is

$$\mathrm{Var}[X] = 1 - \frac{I_1(\kappa)}{I_0(\kappa)} \tag{3.26}$$

The product of two von Mises density functions, the computation of which is needed for performing Recursive Bayesian Estimation, is also (proportional to) a von Mises density function:

$$\begin{aligned} p_{X_1}(\theta) \cdot p_{X_2}(\theta) &\propto \exp\left(\kappa_1 \cos(\theta - \mu_1) + \kappa_2 \cos(\theta - \mu_2)\right) \quad (3.27) \\ &= \exp\left(\hat{\kappa} \cos(\theta - \hat{\mu})\right) \quad (3.28) \end{aligned}$$

where

$$\begin{aligned} \hat{\kappa} &= \sqrt{(\kappa_1 \cos \mu_1 + \kappa_2 \cos \mu_2)^2 + (\kappa_1 \sin \mu_1 + \kappa_2 \sin \mu_2)^2} \quad (3.29) \\ \hat{\mu} &= \operatorname{atan2}\left(\kappa_1 \sin \mu_1 + \kappa_2 \sin \mu_2, \kappa_1 \cos \mu_1 + \kappa_2 \cos \mu_2\right) \quad (3.30) \end{aligned}$$

Care will have to be taken when using the above expressions for the case

$$\begin{bmatrix} \kappa_1 \sin \mu_1 \\ \kappa_1 \cos \mu_1 \end{bmatrix} + \begin{bmatrix} \kappa_2 \sin \mu_2 \\ \kappa_2 \cos \mu_2 \end{bmatrix} = \vec{0} \tag{3.31}$$

for which the angle computed with the atan2$(y, x)$ function is not defined. In such case, the concentration value would be $\hat{\kappa} = 0$, which is the special case of maximum entropy that was discussed above. Given the context where the expressions are used, a meaningful way to handle this scenario may be found.

The convolution of two von Mises density functions is not, however, a von Mises density function. But the result of the convolution can be approximated well by a von Mises density [24], and success has been shown with the use of this approximation in RBE schemes for bearing tracking tasks [25]. That is, the full expression of the p.d.f. of $X = X_1 * X_2$ being

$$p_{X_1 * X_2}(\theta) = \frac{1}{2\pi I_0(\kappa_1) I_0(\kappa_2)} I_0\left(\sqrt{\kappa_1^2 + \kappa_2^2 + 2\kappa_1\kappa_2 + \cos(x - (\mu_1 + \mu_2))}\right) \tag{3.32}$$

can be approximated by the von Mises p.d.f. of a random variable $\hat{X} \sim \text{vonMises}(\hat{\mu}, \hat{\kappa})$ with

$$\begin{aligned} \hat{\mu} &= \mu_1 + \mu_2 \quad (3.33) \\ \hat{\kappa} &= A^{-1}(A(\kappa_1)A(\kappa_2)) \quad (3.34) \\ A(\kappa) &= \frac{I_1(\kappa)}{I_0(\kappa)} \quad (3.35) \end{aligned}$$

This can be helpful if we need to consider process noise, for instance, arising from uncertainty in the rotational movement of the agent, which can be it too modeled with a von Mises distribution, and requires the convolution in the prediction step. In this thesis, this result will not be used, and as such, we will not study the analytical properties of the function defined in Equation (3.34).

We now proceed with the adaptation of the Recursive Bayesian estimator to the 2D task. The system will now require two more states to encode the belief of $\varphi(t)$, and we will denote those new state variables as $\mu_\varphi$ and $K$.

### While in measurement mode

As before, we assume the agent is stationary, due to $q = 0$, and the measurements to be available at time interval $T_m$. In general, please refer to subsection 2.1.4 for the setup and notation details of the RBE information fusion, since only the main differences between the 1D and 2D cases are presented here.

We want to perform RBE of the agent-target distance and bearing, so we introduce the following two p.d.f.s

$$p(d_{t_i} \,|\, \tilde{d}_{t_i}, \dots, \tilde{d}_{t_0}) = \frac{1}{\sqrt{2\pi P_{t_i}}} \exp\left( -\frac{1}{2} \frac{(\mu_{d,\,t_0} - d_{t_i})^2}{P_{t_i}} \right) \tag{3.36}$$

$$p(\varphi_{t_i} \,|\, \tilde{\varphi}_{t_i}, \dots, \tilde{\varphi}_{t_0}, \kappa_{\varphi,\,t_i}, \dots, \kappa_{\varphi,\,t_0}) = \frac{1}{2\pi I_0(K_{t_i})} \exp\left( K_{t_i} \cos(\varphi_{t_i} - \mu_{\varphi,\,t_i}) \right) \tag{3.37}$$

which encode the beliefs at time $t_i$ given the previous measurements with the estimator state variables $\mu_d(t)$, $P(t)$ and $\mu_\varphi(t)$, $K(t)$.

The update of the distance estimate, following a new measurement $\tilde{d}_{t_i}$ at time $t_i$, is performed as before with the maps defined in Equation (2.29). The update of the bearing estimate is instead performed, similarly to the case of the 1D task, by computing the product of the von Mises p.d.f. of the prior (belief at time $t_{i-1}$) and the von Mises p.d.f. of the measurement. We find, starting from the relations of Equation (3.29), the update map for the bearing estimate p.d.f. to be

$$K_{t_{i+1}} = \sqrt{(K_{t_i} \cos \mu_{\varphi,\,t_i} + \kappa_\varphi \cos \tilde{\varphi}_{t_i})^2 + (K_{t_i} \sin \mu_{\varphi,\,t_i} + \kappa_\varphi \sin \tilde{\varphi}_{t_i})^2} \tag{3.38}$$

$$\mu_{\varphi,\,t_{i+1}} = \operatorname{atan2}\left( K_{t_i} \sin \mu_{\varphi,\,t_i} + \kappa_\varphi \sin \tilde{\varphi}_{t_i}, K_{t_i} \cos \mu_{\varphi,\,t_i} + \kappa_\varphi \cos \tilde{\varphi}_{t_i} \right) \tag{3.39}$$

The case outlined before by Equation (3.31) can be easily handled in this context of RBE estimation. If such condition were to hold, then we can set the updated estimate as follows

$$K_{t_{i+1}} = 0 \tag{3.40}$$

$$\mu_{\varphi,\,t_{i+1}} = \mu_{\varphi,\,t_i} \tag{3.41}$$

and thus set the next RBE state to the one of maximum entropy (zero-knowledge). This is also equivalent to performing a reset of the estimator, as it will be clear later when the full hybrid dynamic is described.

In particular, we again have that there exists a "reset state", which is characterized by $K = 0$ and $P \to \infty$, and with the property that a measurement update with such prior belief leads to the estimator state $(\mu_d, P, \mu_\varphi, K)$ being set to the values $(\tilde{d}, \sigma_d^2, \tilde{\varphi}, \kappa_\varphi)$ of the incoming measurement. This can be verified with the above expressions.

### While in movement mode

Since there is no agent actuation uncertainty and the target is assumed to be stationary, as in the 1D case of the previous chapter, the continuous-time update of the distance belief is obtained with the same procedure as before, but with the results previously computed for the 2D setup in Equation (3.9). Then we obtain

$$\dot{\mu}_d(t) = \dot{d}(t)|_{\varphi=\mu_\varphi} \tag{3.42}$$

$$= -\cos(\mu_\varphi(t))v(t) \tag{3.43}$$

where we use the mean $\mu_\varphi(t)$ of the estimate of $\varphi(t)$ instead of the true unknown quantity. As for the variance of the distance estimate, we have

$$\dot{P}(t) = 0 \tag{3.44}$$

as before. From (3.15) we then obtain the continuous-time update equation for the mean of the estimate of $\varphi(t)$ as

$$\dot{\mu}_\varphi(t) = \dot{\varphi}(t)|_{d=\mu_d,\ \varphi=\mu_\varphi} \tag{3.45}$$

$$= \frac{\sin\left(\mu_\varphi(t)\right)}{\mu_d(t)}v(t) - \omega(t) \tag{3.46}$$

where we again used the mean of the estimates of $d(t)$ and $\varphi(t)$ in place of their true unknown value.

Regarding the continuous-time update of the pointing angle estimate concentration $K$, it makes intuitive sense that as we get closer to the estimated target position, the confidence regions of the p.d.f. increase in angular size, and vice versa if the agent is driving away. We find the continuous-time update law for the parameter $K$ as follows: we start from the property that the von Mises density can be approximated by a Gaussian density as outlined in Equation (3.25), under which we have that the variance is $K^{-1}(t)$. Then we compute the arc-length $\ell(t)$ tied to $\gamma > 0$ standard-deviations of the angular error on the circle of radius $\mu_d(t)$ as

$$\ell(t) = \gamma\sqrt{K^{-1}(t)}\mu_d(t) \tag{3.47}$$

and enforce $\dot{\ell}(t) = 0$, so that this quantity stays constant. By neglecting $\gamma$ and differentiating both sides, we find

$$0 = -\frac{1}{2}K^{-\frac{3}{2}}(t)\dot{K}(t)\mu_d(t) + K^{-\frac{1}{2}}(t)\dot{\mu}_d(t) \tag{3.48}$$

then, solving for $\dot{K}(t)$, we obtain

$$\dot{K}(t) = \frac{K^{-\frac{1}{2}}(t)}{\frac{1}{2}K^{-\frac{3}{2}}(t)\mu_d(t)}\dot{\mu}_d(t) = \frac{2K(t)}{\mu_d(t)}\dot{\mu}_d(t) \tag{3.49}$$

$$= -\frac{2K(t)}{\mu_d(t)}\cos(\mu_\varphi(t))v(t) \tag{3.50}$$

where we also performed the substitution of the expression for $\dot{\mu}_d(t)$ obtained above.

### 3.1.4 Control input and movement action

We adopt the control law presented by Siciliano et al. [22, chapter 11.6 "Motion Control"] for Cartesian regulation, once the agent enters movement mode. The equations are rewritten with the quantities we have been designing with, to then obtain the following feedback equations

$$v(t) = \alpha \mu_d(t) \cos(\mu_\varphi(t)) \qquad\qquad \omega(t) = \beta \mu_\varphi(t) \tag{3.51}$$

where, as usual, the mean of the beliefs of the true quantities are used, since their true values are unknown and only estimated through the measurements by the RBE subsystem. The above equations are of clear geometrical interpretation, since the linear velocity is obtained as the projection on the sagittal axis of the agent-target distance vector and the angular velocity is such that the pointing error $\varphi(t)$ converges to zero.

Nevertheless, the above control law is formally proven to make the agent converge to the target position[1] with Lyapunov based arguments and the quadratic function

$$V(d) = \frac{1}{2} d^2 \tag{3.52}$$

along with Barbalat's lemma [22].

### 3.1.5 Two-dimensional switching conditions

To design the mode switching and reset conditions for the 2D control scheme, we recall the geometrical interpretation we presented in subsection 2.1.6, and try to intuitively extend the quantities to the 2D case.

We initially consider the threshold relative to the measurement-to-movement mode switch, which happens when the estimate is considered accurate enough. For such purpose, the quantity $a\sqrt{P}$ was selected, with $a \in \mathbb{R}^+$ being a tunable parameter of the control law, since it represents a distance in meters whose value is proportional to the uncertainty. This positive value was then compared to the absolute value of the current (estimated) distance $|\mu(t)|$.

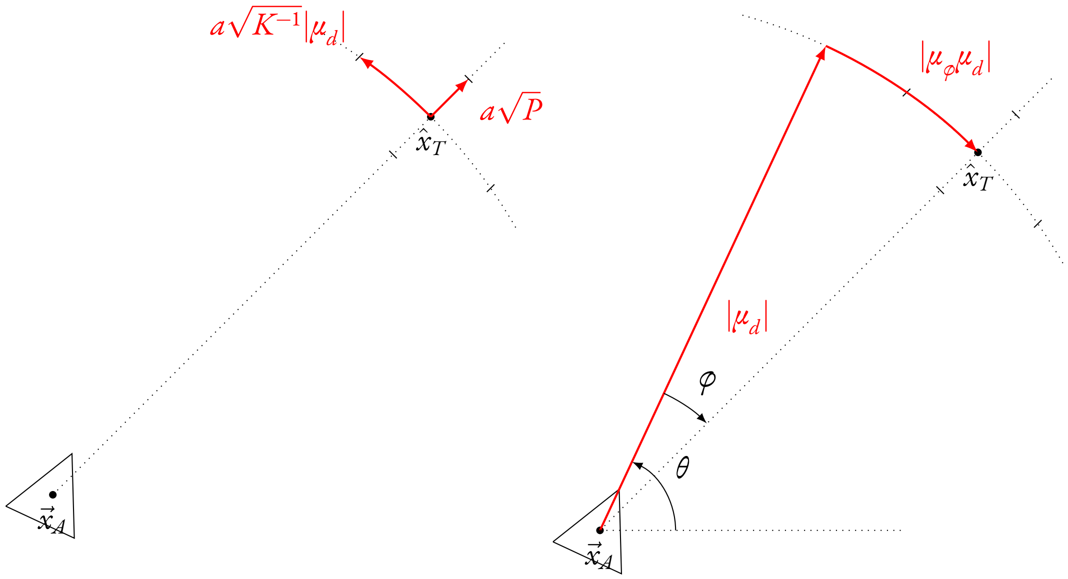
A way to extend those two quantities to the 2D case might be to consider a two-dimensional vector[2] with the two components tied to the two variables being estimated, that is the distance $d(t)$ and the pointing error $\varphi(t)$. Let us then define the following *uncertainty vector*:

$$\vec{s}(t) = \begin{bmatrix} a\sqrt{P(t)} \\ a\sqrt{K^{-1}(t)}|\mu_d(t)| \end{bmatrix} \tag{3.53}$$

where $a \in \mathbb{R}^+$ is a parameter of the control law, as before. The value of the second vector component represents the length of the arc of the circle of radius $\mu_d(t)$ and angle $a\sqrt{K^{-1}(t)}$ and, as such, is a quantity in meters that is tied to the uncertainty of the estimate of $\varphi(t)$. The von Mises Gaussian approximation presented in Equa-

[1] The estimated target position, in this case.

[2] Beware that the space that this *uncertainty vector* belongs to is not to be confused with the 2D plane (and its associated cartesian coordinates vector space) where the task happens, that is the one introduced in subsection 3.1.1. The space, moreover, is a cone, since the two components are positive semi-definite.



**(a)** Length of the components of the vector $\vec{s}$, tied to the uncertainty of the estimation.

**(b)** Length of the components of the Vector $\vec{T}$, tied to the distance to travel and whose length is used as a threshold.

**Figure 3.4:** Geometrical representation of the 2D uncertainty and threshold vectors.

tion (3.25) has been used to obtain a simpler expression with the approximation of the variance for large values of the concentration parameter $\kappa$. Then we introduce a *threshold vector* that, similarly to the 1D case, has its length proportional to the (estimated) distance, and belongs to the positive semidefinite cone of $\mathbb{R}^2$:

$$\vec{T}(t) = \begin{bmatrix} |\mu_d(t)| \\ |\mu_\varphi(t)\mu_d(t)| \end{bmatrix} \tag{3.54}$$

The second component is, similarly to before, obtained by considering the arc length of the circle of radius $\mu_d(t)$ and angle $\mu_\varphi(t)$. Figure 3.4 depicts a geometrical interpretation of these two vectors.

To trigger the jump from the measurement to the movement mode, we then compare the length of these two vectors and wait for the following condition to happen

$$\|\vec{s}(t)\|_2 \leq \|\vec{T}(t)\|_2 \tag{3.55}$$

which, by taking the square on both sides, is equivalent to

$$\|\vec{s}(t)\|_2^2 \leq \|\vec{T}(t)\|_2^2 \tag{3.56}$$

$$a^2 P(t) + a^2 K^{-1}(t)\mu_d^2(t) \leq \mu_d^2(t) + \mu_\varphi^2(t)\mu_d^2(t) \tag{3.57}$$

$$a^2 \left(P(t) + K^{-1}(t)\mu_d^2(t)\right) \leq \mu_d^2(t) \left(1 + \mu_\varphi^2(t)\right) \tag{3.58}$$

Following the intuition developed on the design of the 1D switching conditions, the condition for the movement-to-measurement mode (*reset*) switch is defined as

$$b^2 \left( P(t) + K^{-1}(t)\mu_d^2(t) \right) \geq \mu_d^2(t) \left( 1 + \mu_\varphi^2(t) \right) \tag{3.59}$$

We again manipulate the complementary conditions of equations (3.58) and (3.59) to highlight that for the controller to not switch and keep operating in the current mode, it must hold

$$b^2 < \frac{\mu_d^2(t) \left( 1 + \mu_\varphi^2(t) \right)}{P(t) + K^{-1}(t)\mu_d^2(t)} < a^2 \tag{3.60}$$

and as such, it must again, in the choice of the parameters, hold the following condition:

$$b < a \tag{3.61}$$

since $a, b \in \mathbb{R}^+$.

As a final note, these conditions have been proposed since they experimentally proved to be well performing and simple enough. Various variants may be obtained, for instance by choosing different factors $a_d$, $a_\varphi$ and $b_d$, $b_\varphi$ for obtaining the components of $\vec{s}(t)$. Moreover they are very similar, especially in the tunable parameters, to the ones for the one-dimensional task, and so they seem to represent a good higher-dimensional extension.

A note about an interesting behavior that arises with the choice of using $\|\vec{T}(t)\|$ as a threshold with respect to (multiples of) $\|\vec{s}(t)\|$, instead of just the distance $|\mu_d(t)|$, is made later in subsection 3.2.3.

### 3.1.6 Closed-loop system

The closed-loop SHS for the 2D variant is then formalized similarly to the 1D case in subsection 2.1.7.

The state of the system is defined as

$$\begin{aligned} x &= \begin{bmatrix} \tau & d & \varphi & \mu_d & E & \mu_\varphi & F & q \end{bmatrix}^T && (3.62) \\ &\in \mathcal{X} = [0, T_m] \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \mathbb{R} \times [0,1] \times \mathbb{S}^1 \times [0,1] \times \{0,1\} \subset \mathbb{R}^8 && (3.63) \end{aligned}$$

with the same continuous, invertible mapping from $P$ to $E$ introduced before in Equation (2.69), and the additional mapping

$$F = e^{-K} \in [0,1] \qquad\qquad K = \ln(F^{-1}) \in [0, +\infty) \tag{3.64}$$

with appropriate extensions by continuity as before.

Again, the value $E = 1$ encodes the condition of zero-knowledge regarding the estimate of the agent-target distance $d(t)$, while $E = 0$ translates to perfect knowledge of such quantity. The value of $F = 1$ similarly encodes the condition of zero-knowledge of the pointing error $\varphi(t)$, and $F = 0$ translates to an arbitrarily high concentration of the p.d.f. of the estimate, in fact

$$\lim_{F \to 0^+} \ln(F^{-1}) = +\infty \tag{3.65}$$

$$\ln(F^{-1})|_{F=1} = 0 \tag{3.66}$$

Figure 3.5 plots the continuous invertible mapping.

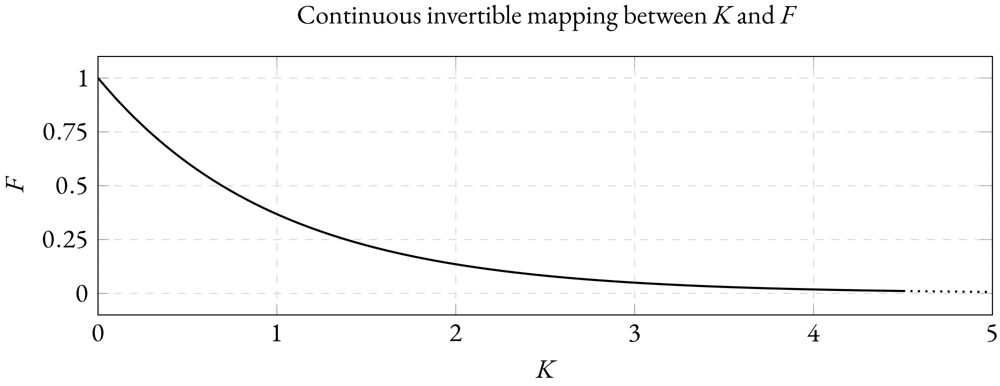


**Figure 3.5:** Plot of the continuous mapping between $K$ and $F$.

We adapt the flow and jump sets as follows:

$$D_m = \{T_m\} \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \{a^2(PK + \mu_d^2) \geq \mu_d^2(1 + \mu_\varphi^2)K\} \times \{0\} \tag{3.67}$$

$$C_0 = [0, T_m] \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \{a^2(PK + \mu_d^2) \geq \mu_d^2(1 + \mu_\varphi^2)K\} \times \{0\} \tag{3.68}$$

$$D_{01} = \{0\} \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \{a^2(PK + \mu_d^2) \leq \mu_d^2(1 + \mu_\varphi^2)K\} \times \{0\} \tag{3.69}$$

$$C_1 = \{0\} \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \{b^2(PK + \mu_d^2) \leq \mu_d^2(1 + \mu_\varphi^2)K\} \times \{1\} \tag{3.70}$$

$$D_{10} = \{0\} \times \mathbb{R} \setminus \{0\} \times \mathbb{S}^1 \times \{b^2(PK + \mu_d^2) \geq \mu_d^2(1 + \mu_\varphi^2)K\} \times \{1\} \tag{3.71}$$

where the thresholds on the estimator variables $\mu_d$, $P$, $\mu_\varphi$, $K$ have been obtained from (3.58) and (3.59) so that they are defined in the entire domain of the variables. Given the mappings from $E$ and $F$ to $P$ and $K$ that are continuous and invertible, with appropriate care, they may be rewritten as a function of them, but for clarity reasons, we currently leave them in this form.

The jump maps are defined as

$$g_m(x, \vec{\nu}) = \begin{bmatrix} 0 & d & \varphi & M_{d,\,E}(\mu_d, E, d + \nu_1) & N_{d,\,E}(E) & M_{\varphi,F}(\mu_\varphi, F, \varphi + \nu_2) & N_{\varphi,F}(F) & 0 \end{bmatrix}^T \tag{3.72}$$

$$g_{01}(x, \vec{\nu}) = \begin{bmatrix} 0 & d & \varphi & \mu_d & E & \mu_\varphi & F & 1 \end{bmatrix}^T \tag{3.73}$$

$$g_{10}(x, \vec{\nu}) = \begin{bmatrix} 0 & d & \varphi & \mu_d & 1 & \mu_\varphi & 1 & 0 \end{bmatrix}^T \tag{3.74}$$

where we used the previous definitions of $M_E$, $N_E$ from Equation (2.82) that have been renamed to $M_{d,\,E}$ and $N_{d,\,E}$ respectively to highlight that they now only refer to the RBE update of the $d$ estimate, and the following new definitions have been used

$$M_\varphi(\mu_\varphi, K, \tilde{\varphi}) := \begin{cases} \operatorname{atan2}\left(K \sin \mu_\varphi + \kappa_\varphi \sin \tilde{\varphi}, K \cos \mu_\varphi + \kappa_\varphi \cos \tilde{\varphi}\right) & \text{if } N_\varphi(\mu_\varphi, K, \tilde{\varphi}) \neq 0 \\ \mu_\varphi & \text{if } N_\varphi(\mu_\varphi, K, \tilde{\varphi}) = 0 \end{cases} \tag{3.75}$$

$$N_\varphi(\mu_\varphi, K, \tilde{\varphi}) := \sqrt{(K \cos \mu_\varphi + \kappa_\varphi \cos \tilde{\varphi})^2 + (K \sin \mu_\varphi + \kappa_\varphi \sin \tilde{\varphi})^2} \tag{3.76}$$

from Equation (3.38), and their counterpart after the change of variables from $K$ to $F$:

$$M_{\varphi,\,F}(\mu_\varphi, F, \tilde{\varphi}) := \begin{cases} M_\varphi(\mu_\varphi, \ln(F^{-1}), \tilde{\varphi}) & \text{if } F \neq 0 \\ \lim_{K \to \infty} M_\varphi(\mu_\varphi, K, \tilde{\varphi}) = \mu_\varphi & \text{if } F = 0 \end{cases} \tag{3.77}$$

$$N_{\varphi,\,F}(\mu_\varphi, F, \tilde{\varphi}) := \begin{cases} \exp\left(-N_\varphi(\mu_\varphi, \ln(F^{-1}), \tilde{\varphi})\right) & \text{if } F \neq 0 \\ \lim_{K \to \infty} \exp\left(-N_\varphi(\mu_\varphi, K, \tilde{\varphi})\right) = 0 & \text{if } F = 0 \end{cases} \tag{3.78}$$

Moreover, $\vec{\nu} \in \mathbb{R}^2$ is now a vector, and its two components refer to the independent measurement noises of the distance and angular sensing, respectively.

Finally, the vector field for the continuous-time flow is

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \tau \\ d \\ \varphi \\ \mu_d \\ E \\ \mu_\varphi \\ F \\ q \end{bmatrix} = f(x) := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} (1 - q) + \begin{bmatrix} 0 \\ -\alpha \cos(\varphi) \cos(\mu_\varphi) \mu_d \\ \alpha \sin(\varphi) \cos(\mu_\varphi) \mu_d / d - \beta \mu_\varphi \\ -\alpha \cos^2(\mu_\varphi) \mu_d \\ \varepsilon \\ \alpha \sin(2\mu_\varphi)/2 - \beta \mu_\varphi \\ 2\alpha F \ln(F^{-1}) \cos^2(\mu_\varphi) \\ 0 \end{bmatrix} q \qquad x \in C_0 \cup C_1 \tag{3.79}$$

where we assume the extension by continuity of the second-to-last component in $F = 0$, that is the full expression would be

$$f_F(x) = \begin{cases} 2\alpha F \ln(F^{-1}) \cos^2(\mu_\varphi) & \text{if } F > 0 \\ 0 & \text{if } F = 0 \end{cases} \tag{3.80}$$

for formal correctness.

The expressions of the vector field above have been obtained from equations (3.9), (3.15), (3.42) and (3.45) by substituting equation (3.51) of the control law. Simplifications have been made whenever possible.

Notice that $d = 0$ is not part of the flow and jump sets, and consequently of the state space, since the change of variables from cartesian to polar coordinates that was introduced in subsection 3.1.1 is not invertible for such value. This is not an issue since, were the value of $d$ to reach 0, the solution would exit the flow and jump sets and stop. However, the condition $d = 0$ coincides with the one of task completion.

Moreover, there is no need for the correction

$$\dot{F}(t) = (\dots) + \varepsilon \tag{3.81}$$

as we instead did for the variable $E$. In fact, by studying the conditions in the definitions of the jump sets of equations (3.67)-(3.70), it is easy to check that the only critical scenario is when $K \to \infty \wedge P = 0$, since with reference to the condition of the sets $C_1$, $D_{10}$ in Equation (3.70):

$$\begin{aligned} \lim_{K\to\infty} \left. \frac{b^2(PK + \mu_d^2)}{K} \right|_{P=0} &= b^2 P\,|_{P=0} && (3.82) \\ &= 0 && (3.83) \\ &\geq \mu_d^2(1+\mu_\varphi^2) \iff \mu_d^2 = 0 \wedge \mu_\varphi^2 = 0 && (3.84) \end{aligned}$$

but the feedback law is not designed to reach such condition in finite time. The system then never switches back to measurement mode with a reset jump. It is enough to ensure that $P$ exits the subspace $P = 0$ for this to not happen, and for the reset condition to trigger as designed. Also note, in this regard, that while it is true that the term $f_F(x)$ is always greater than or equal to zero (since $F\ln(F^{-1}) \geq 0$ and $2\alpha\cos^2(\mu_\varphi) \geq 0$), and it might then seem that this term is enough to make the $F(t)$ variable drift away from the origin, we actually have $F = 0 \implies f_F(x) = 0$, as we indeed made explicit above in Equation (3.80). Thus, the previous reasoning is still important since $F(t)$ has an (unstable) equilibrium in $F = 0$.

It is only left to define the distribution function for the stochastic random jumps, as we did for the 1D system in Equation (2.91):

$$\begin{aligned} &\lambda : \mathbf{B}(\mathbb{R} \times \mathbb{S}^1) \to [0,1] && (3.85) \\ &\omega \mapsto \lambda(\omega) = \int_\omega \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\frac{x_1}{\sigma_d^2}\right) \cdot \frac{1}{2\pi I_0(\kappa_\varphi)} \exp\left(\kappa_\varphi \cos(x_2)\right) \mathrm{d}\vec{x} && (3.86) \end{aligned}$$

obtained by integrating the product of the p.d.f.s of the two *independent* random variables that characterize the distance measurement Gaussian noise, and the angular measurement von Mises noise, on the set $\omega \in \mathbf{B}(\mathbb{R} \times \mathbb{S}^1)$ of the event.

As before, tables 3.1, 3.2 and 3.3 summarise and describe all the state variables, system parameters and controller parameters that characterize the system. The state transition diagram in Figure 2.2, relative to the 1D system, still holds for this proposed 2D extension.

| State variable | Description | Unit |
|---|---|---|
| $\tau$ | Measurement timer, only enabled while $q = 0$ | s |
| $d$ | True agent-target distance | m |
| $\varphi$ | True pointing error | rad |
| $\mu_d$ | Mean of the Gaussian estimate of the agent-target distance | m |
| $P$ | Variance of the Gaussian estimate of the agent-target distance | $\text{m}^2$ |
| $\mu_\varphi$ | Mean of the von Mises estimate of the pointing error | rad |
| $K$ | Concentration of the von Mises estimate of the pointing error | |
| $q$ | Operating mode: $q = 0$ measurement, $q = 1$ movement | |

**Table 3.1:** State variables of the closed-loop SHS for the 2D task.

| Parameter | Description | Unit | Properties |
|---|---|---|---|
| $T_m$ | Time between measurements | s | $> 0$ |
| $\sigma_d^2$ | Variance of the distance measurements | $\text{m}^2$ | $> 0$ |
| $\kappa_\varphi$ | Concentration parameter of the pointing error measurements | | $> 0$ |
| $\varepsilon$ | Positive derivative of $E$ during movement | | $> 0, \ll 1$ |

**Table 3.2:** Parameters of the 2D system.

| Parameter | Description | Unit | Properties |
|---|---|---|---|
| $\alpha$ | Cartesian regulation controller linear velocity gain in movement mode | | $> 0$ |
| $\beta$ | Cartesian regulation controller angular velocity gain in movement mode | | $> 0$ |
| $b$ | Movement to measurement mode switch parameter | | $> 0$ |
| $a$ | Measurement to movement mode switch parameter | | $> b$ |

**Table 3.3:** Parameters of the 2D control law.

It is also worth noticing, looking at equations (3.75), that the von Mises measurement update equations are essentially performing a vector addition, starting from the polar coordinates $(K_{t_{i-1}}, \mu_{\varphi,\, t_{i-1}})$ of the prior vector and the polar coordinates $(\kappa_\varphi, \tilde{\varphi}_{t_i})$ of the measurement vector. Thus, we could equivalently store the cartesian coordinates of the vector associated to the von Mises p.d.f. of the belief, and greatly simplify the jump map expressions. This, however, would come at the cost of more complex expressions for the RBE prediction, which drive the state in continuous-time during the measurement mode, and more complex expressions for the regulation feedback control that takes action in movement mode.

Nevertheless, a final interesting observation regarding the behavior of this proposed 2D variant of the system is that the concentration $K$ of the pointing error $\varphi$ estimate is not monotonically increasing even during the measurement mode. This is easy to understand by considering the above comment relative to the vector representation of the probability density function.

## 3.2 Simulations and behavior

As before, the proposed 2D closed-loop system has been simulated with Algorithm A.1 and its faster variant described by Algorithm A.3 for the Monte Carlo simulations.

### 3.2.1 Behavior in time

We initially show some results regarding the behavior in time of the state variables. As before, we select two different initial conditions, which in this case have been chosen as

$$x_{\text{good}}(0) = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 10 & 1 & -\frac{\pi}{4} & 1 & 0 \end{bmatrix}^T \tag{3.87}$$

$$x_{\text{bad}}(0) = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 2 & 0.01 & +\frac{\pi}{4} & e^{-1000} & 0 \end{bmatrix}^T \tag{3.88}$$

so that we either start from a correct estimate, in the reset condition, or we start with a misleading estimate of $d$ and $\varphi$ to ensure that the system can recover from this condition. Figures 3.6 and 3.7 show the results of these two scenarios.

### 3.2.2 Effect of parameters on task completion metrics

As for the 1D case, we run Monte Carlo simulations to estimate the expected final distance after $t_f = 100$s of operation, for both types of initial conditions, and for the expected time required to reach the target within a radius of $d_R = 0.5$ and with the controller jumping back to measurement mode. The expected values of both metrics are again estimated as a function of the two parameters $a > b > 0$ that characterize the switching behavior. For all simulations, $\alpha = 0.1, \beta = 0.4, \varepsilon = 1 \times 10^{-9}$ and the number of samples for the Monte Carlo estimation is $N = 250$. Figure 3.8 depicts the results for the final distance metric, and Figure 3.9 is relative to the reach time.

Please refer to subsection 2.2.2 for the definitions and details of how these performance metrics are computed.

### 3.2.3 Observations

As before, we discuss interesting behaviors that we observe from the simulation results shown above.

We start with the simulations of the behavior in time of the system of subsection 3.2.1. Figure 3.6 shows the state trajectory in the first case of good initial conditions, and we observe that the system cleanly converges to $d = 0$ and $\varphi = 0$. More in detail, the system switches to movement mode as soon as the first measurement is processed by the RBE estimation, since the condition of Equation (3.58), given the choice of the parameter

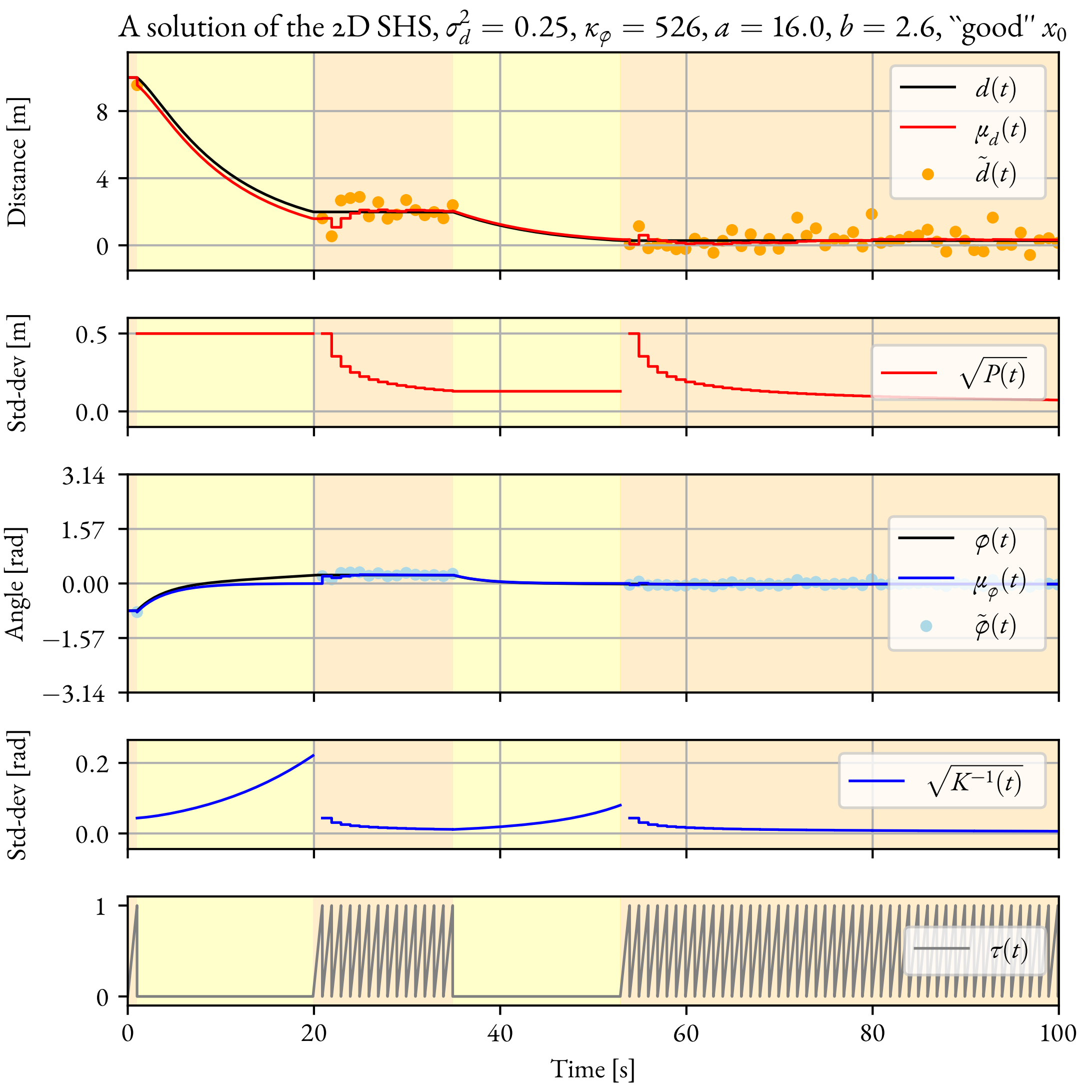


**Figure 3.6:** Behavior in time of the 2D system, obtained by sampling a random solution. All state variables are plotted, with the current mode $q(t)$ being depicted through the differently colored background segments. The parameters are chosen as $\alpha = 0.1$, $\beta = 0.4$, $\varepsilon = 1 \times 10^{-9}$, and $\sigma_d^2, \kappa_\varphi, a, b$ as in the plot title. The initial condition is set to $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 10 & 1 & -\frac{\pi}{4} & 1 & 0 \end{bmatrix}^T$.

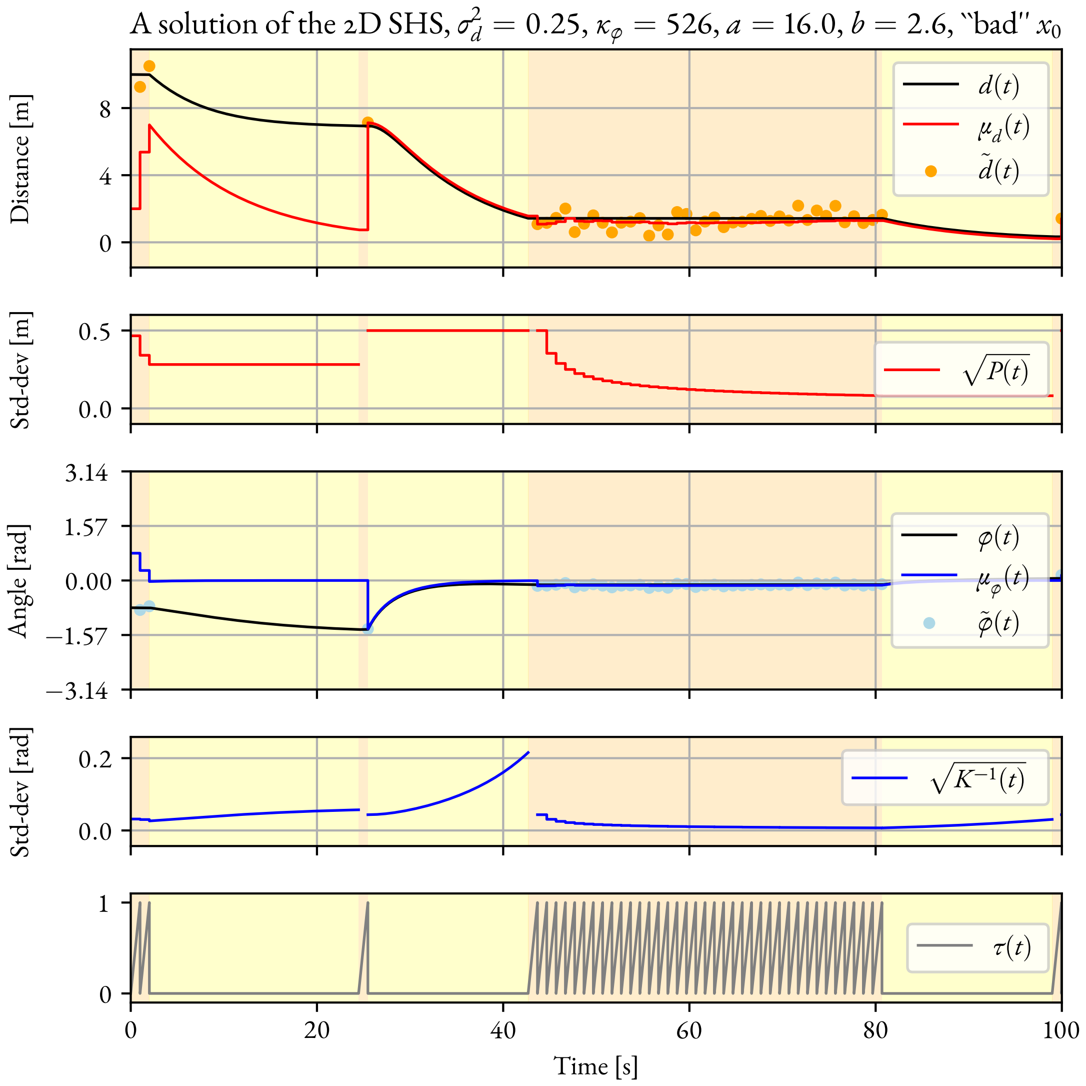


**Figure 3.7:** Behavior in time of the 2D system, obtained by sampling a random solution. All state variables are plotted, with the current mode $q(t)$ being depicted through the differently colored background segments. The parameters are chosen as $\alpha = 0.1,\ \beta = 0.4,\ \varepsilon = 1 \times 10^{-9}$, and $\sigma_d^2, \kappa_\varphi, a, b$ as in the plot title. The initial condition is set to $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 2 & 0.01 & +\frac{\pi}{4} & e^{-1000} & 0 \end{bmatrix}^T$.

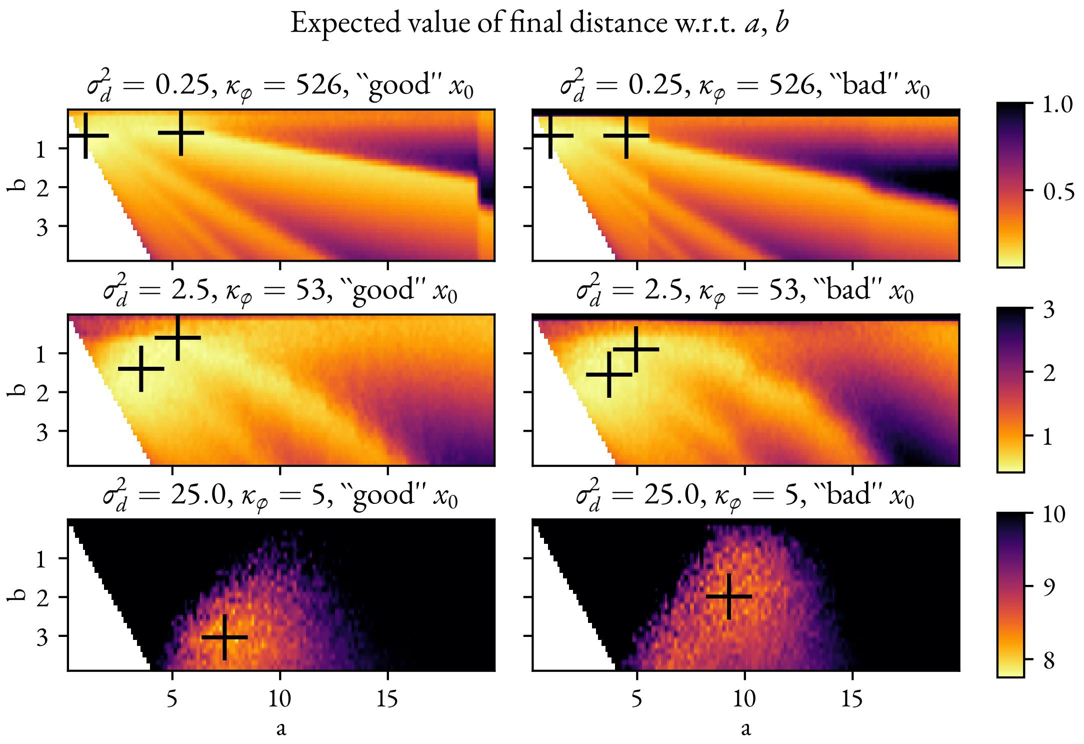


**Figure 3.8:** Expected value of the 2D final distance $\mathbb{E}\left[|d(t_f)|\right]$, with respect to the choice of the two parameters $a, b$, in the presence of low, medium and high measurement noise $\sigma_d^2$ and $\kappa_\varphi$. The final time has been chosen as $t_f = 100\text{s}$ and the parameters as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}, T_m = 1\text{s}$. The ``good'' and ``bad'' $x_0$ cases, with reference to the initial condition, are compared, where the good $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 10 & 1 & -\frac{\pi}{4} & 1 & 0 \end{bmatrix}^T$ and the bad $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 2 & 0.01 & +\frac{\pi}{4} & e^{-1000} & 0 \end{bmatrix}^T$. The crosses denote the location of the local minima in the considered parameter range.

$a = 16$, deems valid to begin the approach of the target even with just the initial accuracy of the estimates, given the estimated agent-target distance. After this first movement phase ends, the system goes back to measurement mode and, this time, waits for a more accurate estimate. This is followed by another movement phase that brings the system very close to the target. The system ends its operation at $t_f = 100\text{s}$ still being in measurement mode, trying to obtain an even more accurate estimate to drive closer, and more precisely, provided enough time.

Figure 3.7 shows the state trajectory in the second case of incorrect initial condition, and we indeed observe that the system ends its first movement phase at $d \simeq 7$ and $\varphi \simeq -\frac{\pi}{2}$. The pointing error angle has even increased in magnitude from its initial value of $\varphi(0) = -\frac{\pi}{4}$. Nevertheless, we observe that the reset mechanism behaves as expected and, after only one measurement, the system switches back to movement mode with a meaningful estimate of the target distance. The controller then behaves similarly to before, switching back and forth between the movement and the measurement phases to decrease the distance to the target as time progresses.

It is also interesting to notice that, while the value of $P(t)$ is constant during the movement phase as in the previous case of the 1D system, the concentration $K(t)$ is now monotonically decreasing in time, due to the continuous-

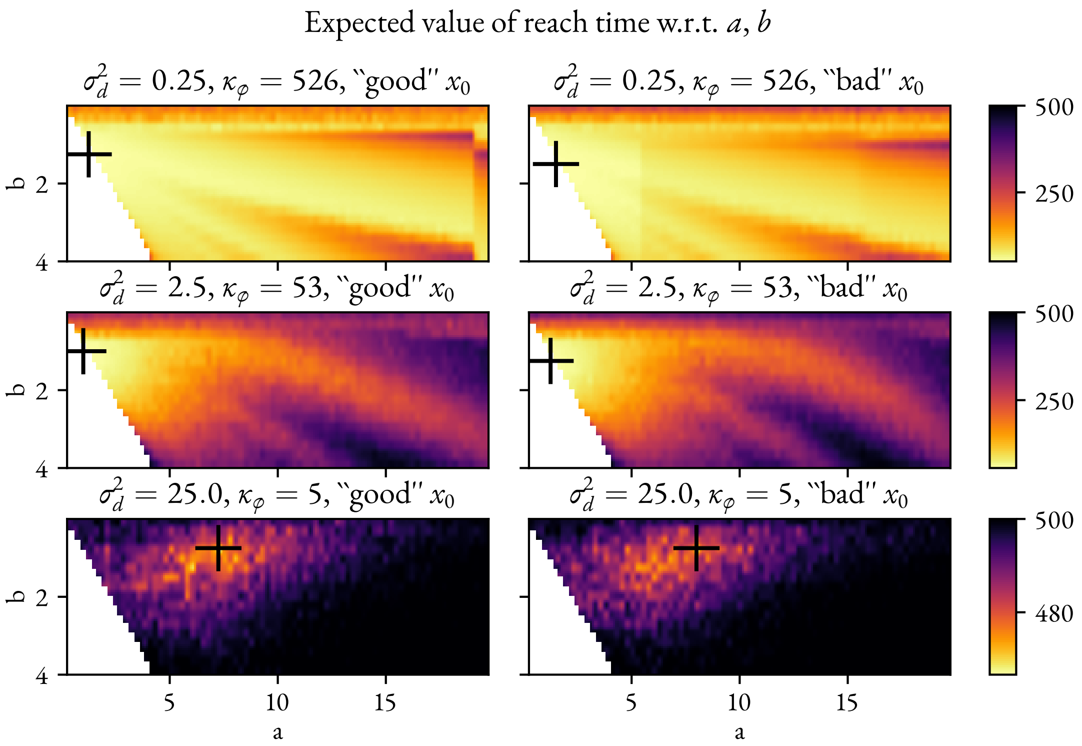


**Figure 3.9:** Expected value of the 2D approach time $\mathbb{E}\left[t_R\right], t_R := \min\{t \geq 0 \,|\, q(t) = 0, |d(t)| \leq d_R\}$, where $d_R = 0.5\text{m}$, with respect to the choice of the two parameters $a, b$, in the presence of low, medium and high measurement noise $\sigma_d^2$ and $\kappa_\varphi$. The maximum operating time is bounded at $500\text{s}$ and the parameters chosen as $\alpha = 0.1, \varepsilon = 1 \times 10^{-9}$ and $T_m = 1\text{s}$. The ``good'' and ``bad'' $x_0$ cases, with reference to the initial condition, are compared, where the good $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 10 & 1 & -\frac{\pi}{4} & 1 & 0 \end{bmatrix}^T$ and the bad $x_0 = \begin{bmatrix} 0 & 10 & -\frac{\pi}{4} & 2 & 0.01 & +\frac{\pi}{4} & e^{-1000} & 0 \end{bmatrix}^T$. The crosses denote the location of the local minima in the considered parameter range.

time update of the belief described by Equation (3.50) and discussed above at the end of subsection 3.1.3. Also in subsection 3.1.6 it was made clear that $\dot{F}(t) \geq 0$, and in particular

$$q = 1 \wedge F \notin \{0, 1\} \wedge \mu_d \notin \left\{\frac{\pi}{2} + \pi k, k \in \mathbb{Z}\right\} \implies \dot{F}(t) > 0 \tag{3.89}$$

which proves the behavior we observe. Figure 3.10 plots the derivative of $F$ as a function of $F$, in its domain, to further highlight its dynamical behavior.

If we finally focus on the first movement-to-measurement mode switch of Figure 3.6, around $t_{s_2} = 20\text{s}$, we observe that the difference between $\varphi$ and $\mu_\varphi$ is increasing in magnitude as the distance gets smaller. This is physically meaningful, and shows that the correction introduced by Equation (3.50) is indeed important to keep a coherent estimator state.

We now focus on the results relative to the performance metrics as a function of the parameter choices.

In general we observe a similar outcome to the 1D case, which was thoroughly discussed in subsection 2.2.3,

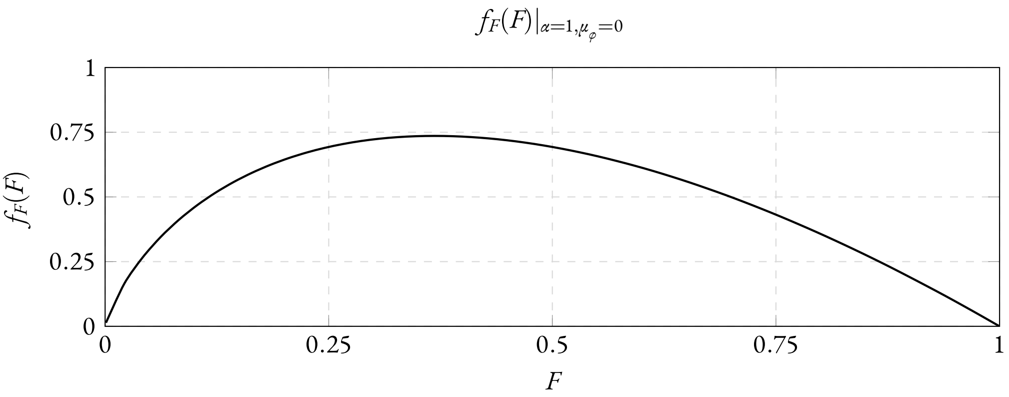


**Figure 3.10:** Plot of the component $f_F(x)$ of the 2D vector field, with respect to $F$ and $\alpha = 1, \mu_\varphi = 0$.

with reference to the presence of two local minima that get closer if the starting initial condition is misleading, and in general move towards higher values of $a$ and $b$ as the measurement noise increases. Actually, this phenomenon can only be observed in the case of the 2D expected final distance depicted in Figure 3.8. We also note that in such experiment, in the specific case of high measurement noise, it is not possible to validate this hypothesis due to the data being very noisy[3]. Finally, we note that this effect is not visible in the experiments related to the reach time shown in Figure 3.9, and we can only observe a single local minima.

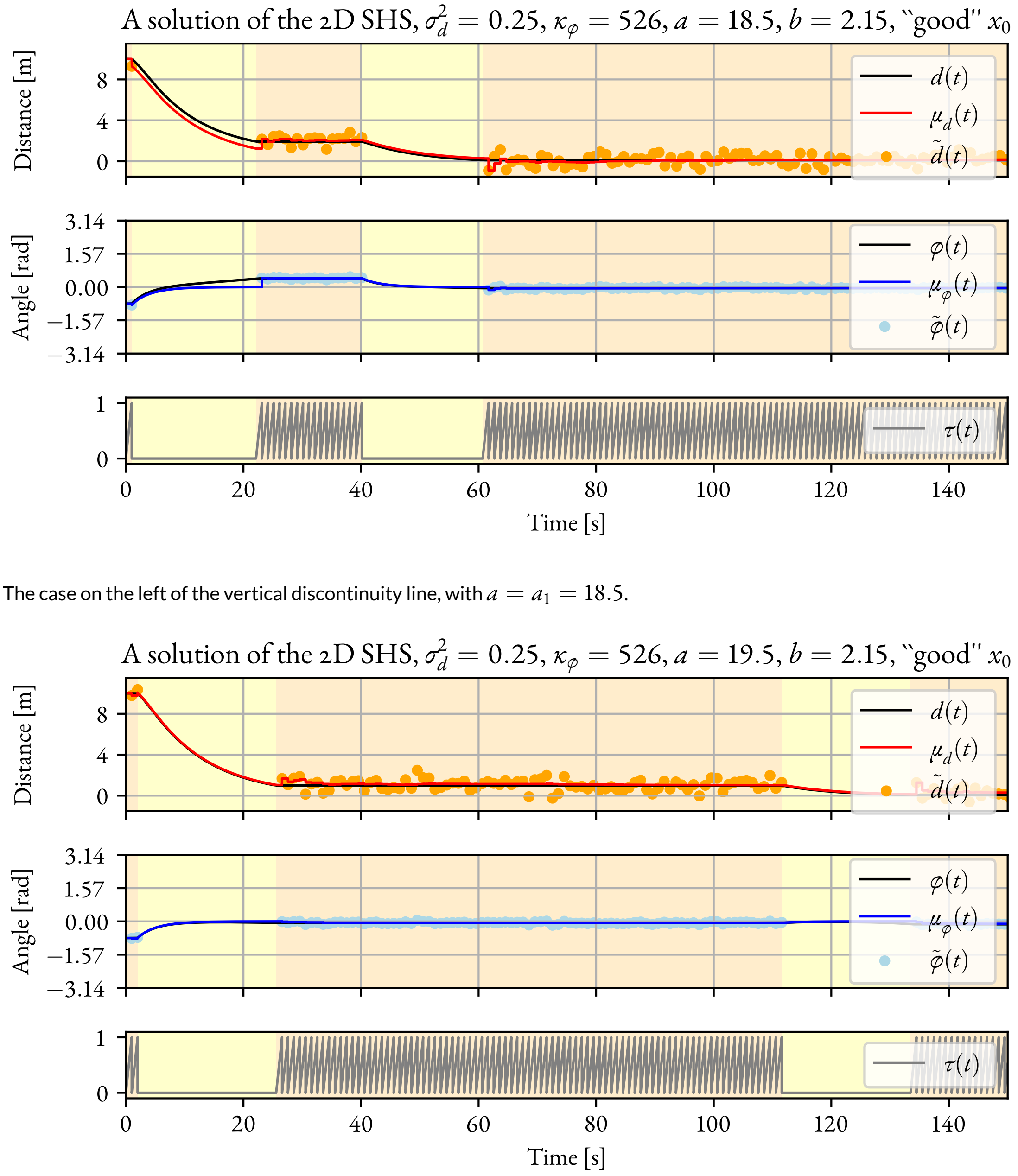
The sharp changes that are visible in the surface plots of figures 3.8 and 3.9, and appear to the eye as vertical lines, have been investigated. To highlight the nature of the phenomenon, we plot two system solutions, in time, that are on two different sides of such jumps in the expected value surface. Without loss of generality, we consider the case of the expected final distance with low measurement noise and favorable initial condition, that is the top-left case of Figure 3.8. We fix the value of $b = 2.15$ and study the cases $a_1 = 18.5$ and $a_2 = 19.5$, which lead to a quite different expected final distance value. Two hybrid arcs relative to those two cases have been sampled and are shown in Figure 3.11a and Figure 3.11b respectively.

We observe that the difference is caused by the system initially performing a different number of measurement updates before entering the first movement phase. In the first case of Figure 3.11a only one measurement is enough for the measurement-to-movement jump condition to be satisfied, while in the second case of Figure 3.11b two measurements are fused together. In particular, this results in a different value of $P$ at the end of the first measurement phase, which stays constant during the movement phase[4] and, thus, makes the movement-to-measurement condition trigger at a different time and at a different agent-target distance. This, in turn, makes the following measurement phase longer, due to the design of the switching conditions, as previously discussed in subsection 2.1.6. This causes the system, at the limit time $t_f = 100$s that characterizes the analysis, to be at two different final distances.

We finally look at the reset jump condition of Equation (3.59), and rewrite it as a function of $\mu_d^2$ in the following

[3] Increasing the number of samples of the Monte Carlo estimation was difficult due to computational reasons.

[4] Without considering the effect of the correction $\dot{E} = \varepsilon$ which is designed to be neglectable.



**(a)** The case on the left of the vertical discontinuity line, with $a = a_1 = 18.5$.

**(b)** The case on the right of the vertical discontinuity line, with $a = a_2 = 19.5$.

**Figure 3.11:** Two random solutions of the 2D system with slightly different values of the parameter $a$, at the left or right side of the discontinuity vertical lines of figures 3.8 and 3.9, that lead to quite different final distances (in expected value).

way:

$$\mu_d^2 \leq \frac{b^2 P}{1 + \mu_\varphi^2 - b^2 K^{-1}} \tag{3.90}$$

for $K > 0$[5]. We notice that the threshold on the squared distance estimate is inversely proportional to the squared pointing error estimate. We might then say that the threshold on the absolute value of the distance increases as the absolute value of the pointing error decreases, and this can be interpreted as a preference of the control law for a zero pointing error to the target. In fact, we might state that the switching threshold accepts a higher agent-target distance for the end of the movement phase, if the pointing error is smaller. If the directional sensing unit of the agent were to perform better when oriented towards the target, that is their performance is anisotropic (like in the case considered by Varotto et. al [3]) then the switching condition that was designed is intuitively also optimizing for this characteristic of the agent. We could then also tune this effect by introducing a multiplicative factor $\psi \geq 0$ for $\mu_\varphi$, which would be equivalent to scaling by $\psi$ the second component of the threshold vector defined in Equation (3.54), obtaining inequalities like

$$b^2(PK + \mu_d^2) \geq \mu_d^2(1 + \psi^2 \mu_\varphi^2)K \tag{3.91}$$

in the case of $D_{10}$ (which we recall is the jump set related to the movement-to-measurement mode switch). In the limit scenario of $\psi = 0$ we remove the effect, and we might then conclude that such case is better suited in the absence of anisotropy of the DOA sensing performance.

In any case, the above reasoning is useful for developing some intuitive understanding of how the switching conditions might be modified to optimize the hybrid control law switching under additional characteristics and limitations of the system.

[5] Otherwise, in its original form of Equation (3.70), for $K = 0$ the outcome is for the condition to trigger since it always holds, $\forall \mu_d \in \mathbb{R}$, that $b^2 \mu_d^2 \geq 0$.

# 4

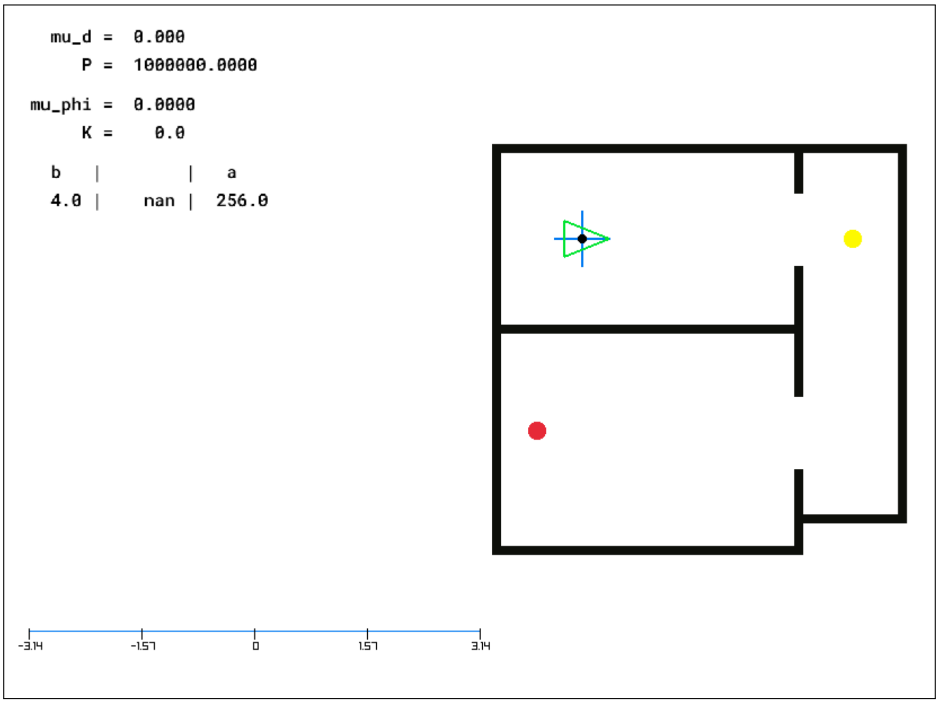
# Localization with sound sensing in unknown environments

As a final experiment, the 2D control law was tested in a simulated Search & Rescue (SAR) scenario under some additional simplifying hypotheses related to the sensing and the navigation. In particular, the peculiarities of sound sensing are exploited, as touched upon in the introductory section 1.3.

An indoor unknown environment is considered, where there are different rooms and corridors. We assume that there exists a path between the agent and the target, which is the victim. Moreover, we assume that the sound waves propagate inside the environment through the corridors and room-to-room or room-to-corridor openings, also along the victim-to-agent path that we assume to exist. We expect the sensing unit to perform some kind of information fusion between the sound DOA estimation data, lidar local mapping data, vision data and/or other sensing modalities, and to provide distance and pointing error measurements $(\tilde{d}(t), \tilde{\varphi}(t))$ that locally represent the location of the perceived sound source in the current room or corridor. For instance, in the initial scenario of Figure 4.1, we expect the sensing unit to measure the location of the sound source at the opening connecting the room to the corridor, due to the following reasons:

- By the Huygens-Fresnel principle, we might state that each opening behaves as a source of spherical waves, like we suppose the real victim to behave as. This can be iteratively applied along the entire victim-agent path.
- The combined audio, visual and/or lidar sensing will detect the opening as the only meaningful local source of the sound. This effectively implements an initial form of navigation inside the unknown environment, with some simple obstacle-avoidance properties.

The simulation is set up for the described scenario, and the pictures in Figure 4.2 are screenshots of the real-time interactive simulator graphical output. The visual elements that are used to depict the simulation state are the following:



```
  mu_d =  0.000
     P =  1000000.0000

mu_phi =  0.0000
     K =    0.0

  b   |         |   a
  4.0 |     nan |  256.0
```

**Figure 4.1:** Snapshot of the unknown environment 2D simulation at the initial state.

- The indoor environment is outlined on the right of the screen, with thick black walls that separate the rooms and the corridor.
- The agent is shown as a green triangle, to highlight the pointing direction, with a black dot in the middle that coincides with its position $\vec{x}_A$ in the plane.
- The victim is shown as a big red circle and is located in the bottom room.
- The currently *locally perceived* location of the victim, which is (in general) different to the real victim location (following the reasons we just outlined), is shown as a yellow circle.
- The last measurement is shown as a small green cross.
- The current estimated target position, obtained from $\mu_d$ and $\mu_\varphi$, is drawn as a big blue cross.
- Two semi-transparent blue circles, centered at the current target estimate, denote (with their radii) the estimated agent-target distances so that
  - if the agent goes outside of the bigger circle, the measurement-to-movement mode switch is triggered;
  - if the agent goes inside of the smaller circle, the movement-to-measurement mode switch is triggered.

  The two circles are drawn only if the estimate is valid, that is $E \neq 1 \wedge F \neq 1$.
- A plot of the current pointing error p.d.f. is shown at the bottom left of the screen.

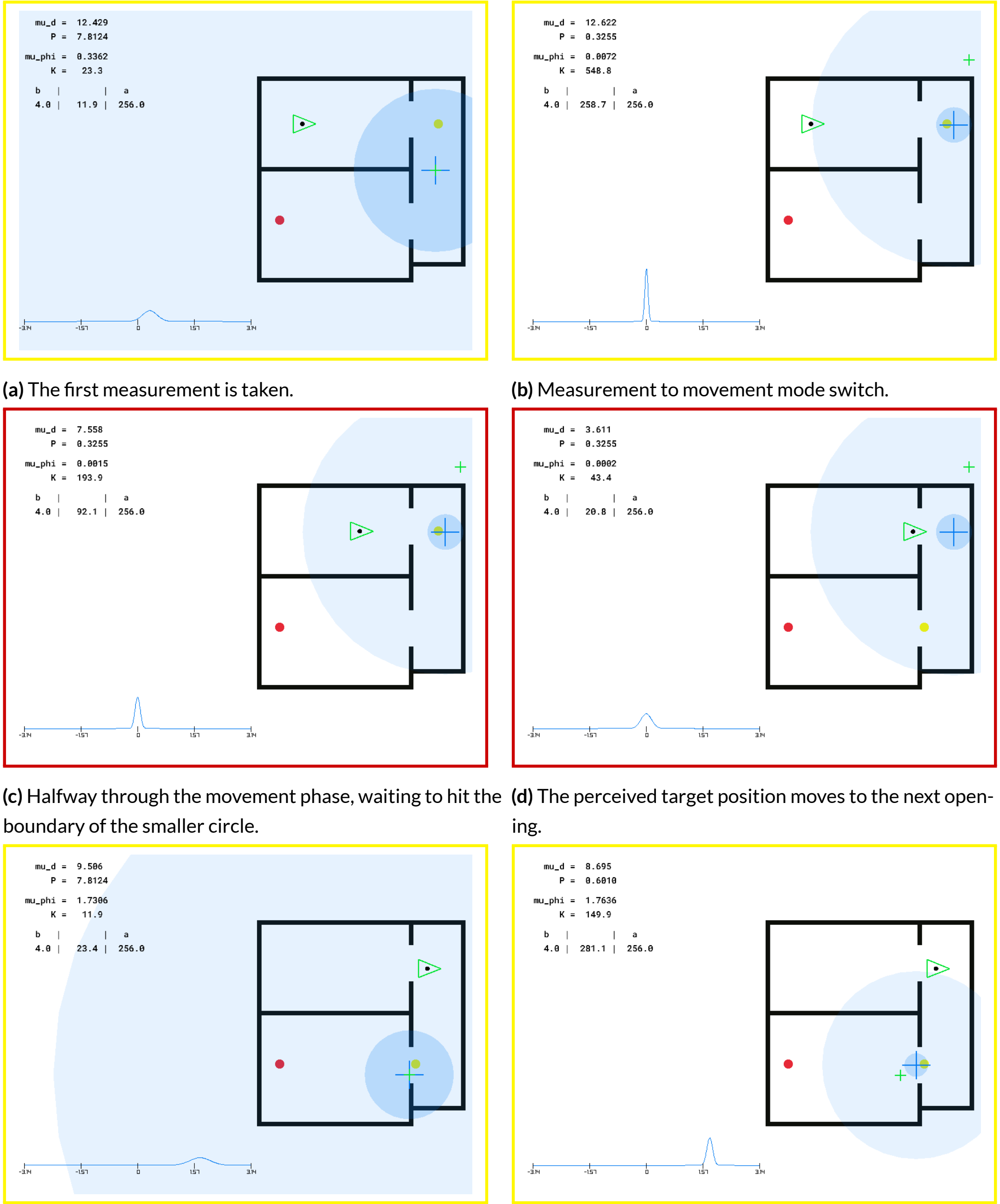
| Parameter | Description | Value | Unit |
|---|---|---|---|
| $T_m$ | Time between measurements | 0.5 | s |
| $\sigma_d^2$ | Variance of the distance measurements | 7.81 | $\mathrm{m}^2$ |
| $\kappa_\varphi$ | Concentration parameter of the pointing error measurements | 19.67 | |
| $\varepsilon$ | Positive derivative of $E$ during movement | $1 \times 10^{-9}$ | |
| $\alpha$ | Cartesian regulation controller linear velocity gain in movement mode | 0.5 | |
| $\beta$ | Cartesian regulation controller angular velocity gain in movement mode | 2.0 | |
| $b$ | Movement to measurement mode switch parameter | 2.0 | |
| $a$ | Measurement to movement mode switch parameter | 16.0 | |

**Table 4.1:** Parameters of the 2D SHS used for the unknown environment simulation.

Figure 4.2 captures noteworthy instants of the simulation state and comments through their progression. A yellow frame denotes a snapshot taken while the measurement mode was active ($q(t) = 0$), while a red frame indicates that the movement mode was active ($q(t) = 1$). So, with the support of the captions relative to the various states shown in the figure, we only briefly comment on the results at a high-level. We observe that the task is completed after three cycles of measurement and movement phases. The first cycle makes the agent go through the room-corridor passage on the top right, which was the initial *perceived* target location. This allows the agent to now perceive the target at the next opening, that is the passage between the bottom room and the corridor (Figure 4.2d). The second cycle of measurement and movement phases brings the agent almost through the second opening (Figure 4.2i). This finally allows the sensing unit to perceive the real victim position, that the agent reaches at the end of this last cycle (Figure 4.2l).

The parameters used for the simulation are listed in Table 4.1, and by considering the values of $\sigma_d^2$ and $\kappa_\varphi$ we might locate the simulated scenario, in terms of magnitude of the measurement noise, as between the medium noise and high noise cases of the experiments in subsection 3.2.2.

We finally notice that the control law does not require the knowledge of the agent state $x_A, y_A, \theta_A$ in the plane. This is not surprising, and given the reasoning at the beginning of this section, we remark this property to highlight that it should be possible, with appropriate tuning and additional work (mainly related to audio-visual information fusion and obstacle avoidance), to exploit the 2D control law designed in this chapter for SAR tasks in unknown environments, without an (indoor) absolute positioning system.



**(a)** The first measurement is taken.

**(b)** Measurement to movement mode switch.

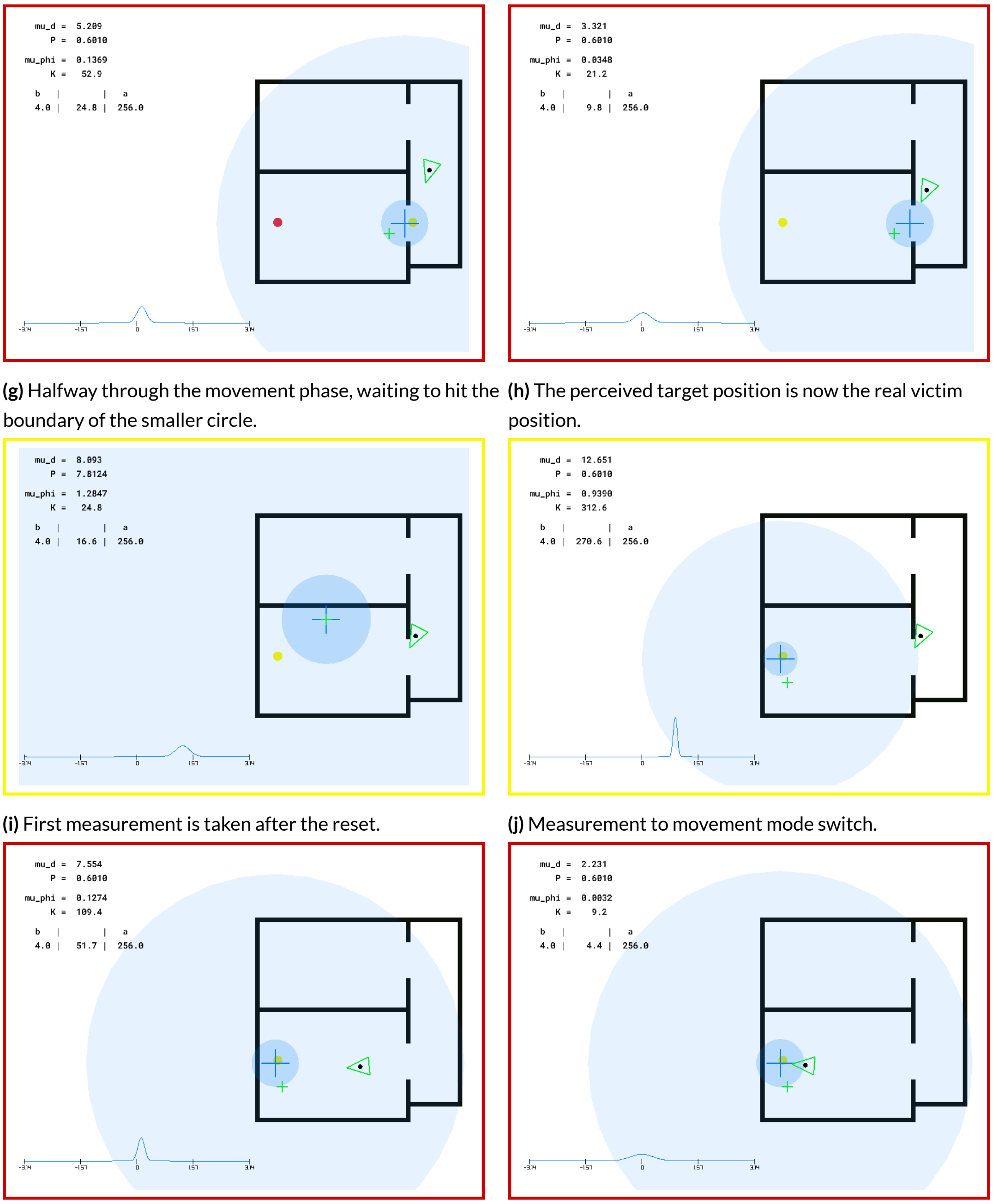
**(c)** Halfway through the movement phase, waiting to hit the boundary of the smaller circle.

**(d)** The perceived target position moves to the next opening.

**(e)** First measurement is taken after the reset.

**(f)** Measurement to movement mode switch.

**Figure 4.2:** Snapshots of the unknown environment 2D simulation at different times.



**(g)** Halfway through the movement phase, waiting to hit the boundary of the smaller circle.

**(h)** The perceived target position is now the real victim position.

**(i)** First measurement is taken after the reset.

**(j)** Measurement to movement mode switch.

**(k)** Halfway through the movement phase, waiting to hit the boundary of the smaller circle.

**(l)** The movement phase is about to end, the perceived position is the real one, and thus the task is completed.

**Figure 4.2:** Snapshots of the unknown environment 2D simulation at different times (continued).

# 5
# Conclusion

## 5.1 Obtained results

Starting from the idea of audio sensing for localization tasks and its peculiarity of being disturbed while the agent actuation is enabled, a SHS in standard form has been obtained in subsection 2.1.7 that models in $\mathbb{R}^1$ the behavior of the proposed mode-switched control law. The system behavior has been studied, both from a theoretical point of view (with the explanations justifying its design in the initial phase, in sections 2.1.2-2.1.6) and from a numerical point of view (with the simulations in time and of the performance metrics of sections 2.2.1 and 2.2.2). The system has been observed to perform qualitatively well at reaching the target under different measurement noise levels.

By adopting the Stochastic Hybrid Systems formalism for modeling, the proposed 1D system of this thesis lays the ground for future work that exploits the tools of SHSs to reason about the behavior and prove properties of the control laws. The design of the system so that the closed-loop is in standard form and satisfies the Stochastic Hybrid Basic Conditions [7] is a first step in this regard. Moreover, the design of the system is such that the subset

$$\mathcal{A} = [0, T_m] \times \{0\} \times \{0\} \times [0,1] \times \{0,1\} \tag{5.1}$$

which we intuitively understand to exhibit some kind of stability property, is compact. This is an useful prerequisite for various results which may be useful in this context [5] [7].

The 2D case has then been considered and a suitable control law obtained, starting from the one-dimensional design. Insight is given in the 2D case on how to adapt (or design from scratch) switching thresholds that lead to different behaviors and might be more or less optimal under different scenarios and specifications.

Finally, the 2D control law has been tested in a simulated environment for a Search & Rescue mission with an appropriately modeled simplified audio-visual sensing device to show the potential applications of the work extended to the two-dimensional plane.

In conclusion, we might state that, in the space of all control laws for such a control task, a class of Stochastic Hybrid control laws is found, with its switching behavior being parametrised by two positive scalars ($a$, $b$), and its movement behavior being parametrised by an appropriate set of gains ($\alpha$, $\beta$) (or just $\alpha$, depending on the dimensionality of the problem). Neglecting the gains $\alpha$ and $\beta$ associated to the cartesian regulation law, whose tuning we may argue only depends on the agent actuation, this allows to optimize the performance in a smaller space, more specifically

$$\{a > b > 0 \,|\, a \in \mathbb{R}, b \in \mathbb{R}\} \subset \mathbb{R}^2 \tag{5.2}$$

which then makes computationally easier to search for an optimal law for a given system and task specifications.

## 5.2 Possible extensions

During the initial research phase, various extensions of the work have been identified, which are briefly described here as a reference for possible future work.

### 5.2.1 Movement uncertainty

The agent actuation has been assumed without noise, and the victim dynamic to be constant. An extension might include the process noise arising from the actuation uncertainty of the agent and/or the random walking of the target. This may only require the RBE to account for the increase of the estimate variance.

### 5.2.2 Measurement noise dependent on distance

The measurement noise has been assumed constant, but it might be interesting to model it as a function of the agent-target distance. In the one-dimensional case, for instance, we might assume that there exists some positive constant $c$ such that

$$\sigma_d^2(x) \leq c|d| \tag{5.3}$$

In the two-dimensional case, we might model some kind of relation between the pointing error measurement uncertainty and the agent-target distance, which might be due to the specific implementation of DOA estimation and/or information fusion performed by the sensing unit.

### 5.2.3 Intermittently detectable target

We might assume that the target is not always detectable. That is, it is not enough for the agent to disable the actuation for the detection to happen, but there is also an hidden state of the target (which might be modeled as an additional boolean logic variable) which determines whether its position can be measured or not.

We may hypothesize, for instance, that this hidden logic variable is driven by a Hybrid System so that it switches in a specific pattern. The control law might then be modified to also account for this behavior, and try to schedule the measurement phases so that they happen only when the target is detectable.

An additional challenge might be to model the detectable state of the target as being driven by a Markov Jump Process, with its transitions happening at random times (which are characterized by specific random variables). The controller might then try to predict the detectability changes of the target to, as in the previous example, synchronize the measurement phase with the intervals during which the target is detectable.

### 5.2.4 Multi-agent and multi-target scenario

We might consider the presence of more than one target, which would then require the agent to prioritize the detections with some appropriate criteria.

Finally, a multi-agent setup might also be considered, where the agents might detect themselves as candidate targets (with audio sensing, for instance, the movement of one agent will produce noise that is detected by other agents and interpreted as a target). Assuming some form of communication, some strategy might be considered to avoid interference and achieve some form of (optimal) coordination between the agents.

# A

# Appendix

## A.1 1D SHS properties

For the SHS designed in section 2.1, we formally prove some properties of its data.

First off, we consider the following description of the sets, where the extensions by continuity have been made explicit taking care of critical values of $\mu$ and $E$. Moreover, we rewrite the inequalities so that that the thresholds for $D_{10}$ and $C_1$ are relative to the estimate $\mu$ (which is the quantity driving the movement, and is strictly decreasing):

$$D_m = \{T_m\} \times \mathbb{R} \times (\{\mu \neq 0, E \geq e^{-a^2/\mu^2}\} \cup \{\mu = 0\}) \times \{0\} \tag{1}$$

$$D_{01} = \{0\} \times \mathbb{R} \times (\{\mu \neq 0, E \leq e^{-a^2/\mu^2}\} \cup \{E = 0, \mu = 0\}) \times \{0\} \tag{2}$$

$$D_{10} = \{0\} \times \mathbb{R} \times (\{E \neq 0, E \neq 1, \mu^2 \leq b^2/\ln(E^{-1})\} \cup \{E = 0, \mu = 0\} \cup \{E = 1\}) \times \{1\} \tag{3}$$

$$C_0 = [0, T_m] \times \mathbb{R} \times (\{\mu \neq 0, E \geq e^{-a^2/\mu^2}\} \cup \{\mu = 0\}) \times \{0\} \tag{4}$$

$$C_1 = \{0\} \times \mathbb{R} \times (\{E \neq 0, E \neq 1, \mu^2 \geq b^2/\ln(E^{-1})\} \cup \{E = 0\}) \times \{1\} \tag{5}$$

We can then prove the following

**Proposition A.3** *The sets $D_m$, $D_{01}$, $D_{10}$, $C_0$ and $C_1$ are closed in $\mathbb{R}^5$.*

Proof We only prove the property for $C_0$, since the closedness of the other sets can be proven in a similar way.

Consider the set $\Sigma$ and its following equivalent definitions

$$\begin{aligned}\Sigma &= \{\mu, E \,|\, \mu \neq 0, E \geq e^{-a^2/\mu^2}\} \cup \{\mu, E \,|\, \mu = 0\} && (6)\\ &= \{\mu, E \,|\, f(\mu, E) \geq 0\} && (7)\\ &= f^{-1}([0, +\infty)) && (8)\\ &\subset \mathbb{R} \times [0, 1]\end{aligned}$$

given $a > 0$ and introducing the function $f\colon\ \mathbb{R} \times [0, 1] \to \mathbb{R}$ as

$$f(\mu, E) = \begin{cases} E - e^{-a^2/\mu^2} & \text{if } \mu \neq 0 \\ E & \text{if } \mu = 0 \end{cases} \tag{9}$$

Notice that $f(\mu, E)$ is a continuous function. In particular $\mu \mapsto e^{-a^2/\mu^2}$ is continuous in $\mathbb{R} \setminus \{0\}$, being obtainable by the function composition of continuous functions. Moreover

$$\lim_{\mu \to 0^+} E - e^{-a^2/\mu^2} = E \quad \forall E \in [0, 1] \tag{10}$$

We can then highlight that $\Sigma = f^{-1}([0, +\infty))$ is a closed set, being the preimage of a closed set through a continuous function. Finally we have that $C_0 = [0, T_m] \times \mathbb{R} \times \Sigma \times 0$ is the cartesian product of closed sets and, as such, is closed. ■

## A.2 SHS solutions generation algorithms

Algorithm A.1 is used to sample random solutions of the stochastic hybrid systems presented in this thesis. A list of samples of the hybrid arc of the solution [5] is returned, and the algorithm takes as parameters

- the initial condition $x_0 \in \mathbb{R}^n$,
- the flow vector field $f\colon \mathbb{R}^n \to \mathbb{R}^n$,
- the flow set $C \subset \mathbb{R}^n$,
- pairs of jump sets and maps $J = \{(D_i, g_i(\cdot))\}$,
- a random variable $\lambda$ to be sampled during jumps,
- a termination condition $T(\cdot)$ which is function of the current hybrid time $(t, j)$ and the current state $x(t, j)$.

The algorithm exploits the procedure A.2 that simulates the continuous-time deterministic flow of the system for a small time interval dt while checking that the state is still inside of the given flow set $C$. If the procedure detects that this is not true, even in the middle of the flow, it returns earlier with a possibly shorter (in time) flow result. This is made possible by the implementation details of the Runge-Kutta method, which has been selected

**Algorithm A.1** STOCHASTIC_HYBRID_SIMULATE($x_0, f(\cdot), C, J, \lambda$, dt, $T(\cdot)$)

```
$x \leftarrow x_0$
$t \leftarrow 0$
$j \leftarrow 0$
result $\leftarrow$ [ ]
result.append(($t, j$, None, $x$))
while not $T(t, j, x)$
    jumped $\leftarrow$ false
    for all $(D_i, g_i(\cdot)) \in J$
        if $x \in D_i$
            $\nu \leftarrow$ SAMPLE($\lambda$)
            $x \leftarrow g_i(x, \nu)$
            $j \leftarrow j + 1$
            result.append(($t, j, \nu, x$))
            jumped $\leftarrow$ true
        end if
    end for
    maybe_flowed_to $\leftarrow$ RK2_FLOW_INSIDE($x, f, C$, dt)
    if maybe_flowed_to is Some(dt$_f$, $x_f$)
        $x \leftarrow x_f$
        $t \leftarrow t +$ dt$_f$
        result.append(($t, j$, None, $x$))
    else
        if jumped
            continue # Couldn't flow, but jumped
        else
            break # Couldn't flow, didn't jump
        end if
    end if
end while
return result
```

**Procedure A.2** RK2_FLOW_INSIDE($x_0, f(\cdot), C$, dt)

```
if $x_0 \notin C$
    return None
end if
$x_1 \leftarrow x_0 + f(x_0)\left(\frac{1}{2}\,\mathrm{dt}\right)$
if $x_1 \notin C$
    return Some $\left(\frac{1}{2}\,\mathrm{dt}, x_1\right)$
end if
$x_2 \leftarrow x_0 + f(x_1)\,\mathrm{dt}$
return Some $(\mathrm{dt}, x_2)$
```

of order 2, and as such is also known as the Midpoint method. This feature of the flow procedure is exploited by Algorithm A.1 to ensure the correct generation of solutions whose hybrid time domain is bounded [5].

Please note that, in the actual implementation and usage of the algorithm, some slack is expected in the definitions of the flow and jump sets. This ensures proper behavior of the algorithm without resorting to very small values of dt which would otherwise make the execution very computationally expensive.

In particular, when the slack in the sets definitions is enough, and the system data is sufficiently well-behaved[1], a faster version of the algorithm can be used. This variant is described by Algorithm A.3, together with its helper procedure A.4. The differences are that the faster version doesn't check for the state to be inside *C* when flowing, and only allows for multiple jumps at the same *t* if their ``instructions'' are checked in the correct sequential order in the collection *J*.

**Algorithm A.3** STOCHASTIC_HYBRID_SIMULATE_FAST($x_0$,$f(\cdot)$,$J$, $\lambda$, dt, $T(\cdot)$)

```
$x \leftarrow x_0$
$t \leftarrow 0$
$j \leftarrow 0$
result $\leftarrow$ [ ]
result.append(($t$,$j$, None, $x$))
while not $T(t,j,x)$
    for all $(D_i, g_i(\cdot)) \in J$
        if $x \in D_i$
            $\nu \leftarrow$ SAMPLE($\lambda$)
            $x \leftarrow g_i(x,\nu)$
            $j \leftarrow j+1$
            result.append(($t$,$j$, $\nu$, $x$))
        end if
    end for
    $x \leftarrow$ RK2_FLOW($x$,$f$, dt)
    $t \leftarrow t +$ dt
    result.append(($t$,$j$, None, $x$))
end while
return result
```

**Procedure A.4** RK2_FLOW($x_0$,$f(\cdot)$, dt)

```
$x_1 \leftarrow x_0 + f(x_0)\left(\frac{1}{2}\,\mathrm{dt}\right)$
$x_2 \leftarrow x_0 + f(x_1)\,\mathrm{dt}$
return $x_2$
```

With care for the limitations that were just described, this variant has proven very useful and better performing (likely thanks to a lower stress on the CPU branch predictor unit) for running a high amount of simulations of the systems presented in this work.

[1] Those characterizations, with respect to the system data, are given in a qualitative way, and asserting them has been mainly performed experimentally.

**Listing A.1** Stochastic Hybrid solutions generation algorithm implementation in Rust.

```
use rand::Rng;

pub fn hybrid_sim<
    const N: usize,                             // dimension of the state space
    const K: usize,                             // number of jump sets and maps
    JS: Fn([f64; N], &P) -> bool,               // type of the jump sets
    JM: Fn([f64; N], f64, &P) -> [f64; N],      // type of the jump maps
    FS: Fn([f64; N], &P) -> bool,               // type of the flow set
    F: Fn([f64; N], &P) -> [f64; N],            // type of the flow vector field
    D: rand::distributions::Distribution<f64>,  // distribution for random jumps
    T: Fn(f64, u64, [f64; N], &P) -> bool,      // type of the termination condition
    P,                                          // type of the user provided system parameters
>(
  x0: [f64; N],
  p: &P, f: F, flow_set: FS, jumps: [(JS, JM); K], lambda: D,
  dt: f64, terminate: T
) -> Vec<(f64, u64, Option<f64>, [f64; N])> {
    let mut x = x0;
    let mut (t, j) = (0.0, 0);
    let mut rng = rand::thread_rng();
    let mut res = vec![(t, j, None, x)];

    while !terminate(t, j, x, p) {
        let mut jumped = false; // check for jumps
        for (jump_set, jump_map) in &jumps {
            if jump_set(x, p) {
                let nu = rng.sample(&lambda);
                x = jump_map(x, nu, p);
                j += 1;
                res.push((t, j, Some(nu), x));
                jumped = true;
            }
        }

        #[cfg(debug)] // in debug mode we use the complete version with checks
        {
            match rk2_flow_inside(x, p, &f, &flow_set, dt) {
                Some((fdt, xf)) => {
                    x = xf;
                    t = t + fdt;
                    res.push((t, j, None, x));
                }
                None if jumped => continue; // couldn't flow, but jumped
                None if !jumped => break;   // couldn't flow, didn't jump
            }
        }
        #[cfg(not(debug))] // in release mode we use the faster version
        {
            x = rk2_flow(x, p, &f, dt);
            t = t + dt;
            res.push((t, j, None, x));
        }
    }

    return res;
}
```

The algorithms have been implemented in the Rust programming language, which offers a highly optimizing AOT compiler, leading to a high performance simulator which allows to quickly generate a big number of solutions. This is critical for Monte Carlo experiments. The implementation of the algorithms is shown in listing A.1 and exploits generics and trait bounds so that the compiler can specialize the code for the specific system being simulated, which is identified by the provided jump sets, jump maps, flow set, flow map and random distribution. The implementation also allows to provide custom parameters for the system, as a reference to a value of a generic type `P` which is accepted by the function.

# References

[1] L. Varotto, A. Cenedese, and A. Cavallaro, ``Active sensing for search and tracking: A review,'' *ArXiv*, vol. abs/2112.02381, 2021.

[2] J. Radak, L. Baulig, D. Bijak, C. Schowalter, and H. Frey, ``Moving towards wireless sensors using rssi measurements and particle filtering,'' in *Proceedings of the 14th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ser. PE-WASUN '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 33–40.

[3] L. Varotto and A. Cenedese, ``Transmitter discovery through radio-visual probabilistic active sensing,'' in *2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2021, pp. 191--196.

[4] M. Hutchinson, C. Liu, P. Thomas, and W.-H. Chen, ``Unmanned aerial vehicle-based hazardous materials response: Information-theoretic hazardous source search and reconstruction,'' *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 108--119, 2020.

[5] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 03 2012.

[6] A. R. Teel, A. Subbaraman, and A. Sferlazza, ``Stability analysis for stochastic hybrid systems: A survey,'' *Automatica*, vol. 50, no. 10, pp. 2435--2456, 2014.

[7] A. R. Teel, ``Lyapunov conditions certifying stability and recurrence for a class of stochastic hybrid systems,'' *Annual Reviews in Control*, vol. 37, no. 1, pp. 1--24, 2013.

[8] J. Cech, R. Mittal, A. Deleforge, J. Sanchez-Riera, X. Alameda-Pineda, and R. Horaud, ``Active-speaker detection and localization with microphones and cameras embedded into a robotic head,'' in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 203--210.

[9] J.-M. Valin, F. Michaud, J. Rouat, and D. Letourneau, ``Robust sound source localization using a microphone array on a mobile robot,'' in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, 2003, pp. 1228--1233 vol.2.

[10] X. Zhong, L. Sun, and W. Yost, ``Active binaural localization of multiple sound sources,'' *Robotics and Autonomous Systems*, vol. 85, pp. 83--92, 2016.

[11] A. Ishfaque and B. Kim, ``Real-time sound source localization in robots using fly ormia ochracea inspired mems directional microphone,'' *IEEE Sensors Letters*, vol. 7, no. 1, pp. 1--4, 2023.

[12] A. Deleforge, D. Di Carlo, M. Strauss, R. Serizel, and L. Marcenaro, ``Audio-based search and rescue with a drone: Highlights from the ieee signal processing cup 2019 student competition [sp competitions],'' *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 138--144, 2019.

[13] D. Bush and N. Xiang, ``A model-based Bayesian framework for sound source enumeration and direction of arrival estimation using a coprime microphone arraya)," *The Journal of the Acoustical Society of America*, vol. 143, no. 6, pp. 3934--3945, 06 2018.

[14] F. Dong, Y. Jiang, Y. Yan, Q. Yang, L. Xu, and X. Xie, ``Direction-of-arrival tracking using a co-prime microphone array: A particle filter perspective," *Applied Acoustics*, vol. 170, p. 107499, 2020.

[15] K. Nakadai, T. Matsui, H. Okuno, and H. Kitano, ``Active audition system and humanoid exterior design," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 2, 2000, pp. 1453--1461 vol.2.

[16] T. Yoshida and K. Nakadai, ``Active audio-visual integration for voice activity detection based on a causal bayesian network," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 370--375.

[17] O. Mokrenko, C. Albea-Sánchez, L. Zaccarian, and S. Lesecq, ``Feedback scheduling of sensor network activity using a hybrid dynamical systems approach," *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 1595--1600, 2015.

[18] C. Albea-Sánchez, A. Seuret, and L. Zaccarian, ``Hybrid control of a three-agent network cluster," *53rd IEEE Conference on Decision and Control*, pp. 5302--5307, 2014.

[19] P. D. Lewis F.L., Xie L., *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. CRC Press, 2008.

[20] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, ``A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174--188, 2002.

[21] G. Pavliotis, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, ser. Texts in Applied Mathematics. Springer New York, 2014.

[22] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2010.

[23] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley & Sons, 2011.

[24] S. R. Jammalamadaka, A. Sengupta, and A. Sengupta, *Topics in circular statistics*. World Scientific, 2001, vol. 5.

[25] I. Marković and I. Petrović, ``Bearing-only tracking with a mixture of von mises distributions," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 707--712.

# Acknowledgments

Voglio ringraziare i miei relatori per il prezioso tempo che mi hanno dedicato in questi mesi. In particolare il professor Cenedese per la sua pazienza e disponibilità anche oltre i fini della tesi, e il professor Zaccarian per l'ospitalità presso UNITN, nonchè per l'aiuto e gli insegnamenti che sono stati fondamentali per il lavoro. Ringrazio anche i compagni di università che mi hanno accompagnato in questo percorso. In particolare Federico, Francesco e Marco per aver da sempre creato un'ambiente stimolante in cui mettermi in gioco, come studente e come persona. Grazie alla mia famiglia e ad Alessia per il supporto, l'ascolto e la vicinanza in questi anni. Un grazie infine a tutti gli amici, di una vita e non, che mi hanno sostenuto per arrivare a questo traguardo.