



Perception-Driven Tube Scaling for State-Dependent Dynamic Tube-Based MPC
Application to disturbance-aware search and rescue scenarios

Filip Keerberg¹

Supervisors: A. Jamshidnejad¹, S. Schoonebeek¹

¹EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Filip Keerberg
Final project course: CSE3000 Research Project
Thesis committee: A. Jamshidnejad, S. Schoonebeek, L. C. Siebert

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Effective autonomous search and rescue (SaR) requires safe and efficient traversal of hazardous regions. While model predictive control (MPC) offers a solid framework for controlling an autonomous agent, the robust tube MPC (TMPC) approach sacrifices too much performance, especially where disturbance probabilities vary spatially. Current approaches to address this problem explore varying tube size in scenarios with perfect information, but none test whether local perception of the environment can be used to dynamically scale tube size to reduce conservativeness adaptively. A state-dependent dynamic TMPC (SDD-TMPC) formulation is proposed, scaling tube dynamically on perceived disturbance risk and confidence. Three controllers: MPC, TMPC, SDD-TMPC, are tested over 250 seeded SaR simulations, with additional runs introducing noise into perceived disturbance risk to assess the formulation's robustness. SDD-TMPC is shown to substantially outperform TMPC (odds-ratio (OR)=40.0) while maintaining 0 wall/victim collisions, operating in TMPC mode only 32.3% of the time. Robustness and performance advantage are retained under perceived noise of magnitudes up to 0.2, however, 0.4 is identified as the boundary where this no longer holds. Overall, this suggests that perception-based tube scaling is a promising direction for autonomous agents.

1 Introduction

Post-disaster environments, such as natural disaster areas, building collapses, and industrial accidents, can become incredibly hazardous settings. Conducting search and rescue (SaR) operations in these areas can expose responders to many physical risks. Because of these risks to human life, research has focused on utilizing robotics for SaR in such environments. Liu et al. [1] discuss the ongoing research of SaR robotics, as well as identify key performances of these models: survivability, mobility, sensing, communicability and operability. SaR robots have been shown to be flexible, and provide value and utility in urban search and rescue, as well as military applications [2].

SaR is a time-critical endeavor, where the speed and performance of the responders can directly influence the chance of victims' survival. The rate of survival of victims has been shown to decrease drastically with each passing day, where the first 72 hours are the most important [1]. A further empirical link can clearly be seen in [3], where they conclude that time is the single most valuable predictor for determining the probability of finding survivors.

Model predictive control (MPC) is a promising framework for SaR robotics, as it can optimize for multiple objectives simultaneously and systematically. For SaR, this could be leading victims to safety, while avoiding hazards along the way. The most noteworthy hazard for the following paper is external hazards that introduce state disturbances, which is when

the position of the robot is altered by the environment. MPC has been shown to be robust to minor disturbances and uncertainties, however, fails when larger external disturbances are introduced [4].

To counteract this limitation, a disturbance-invariant 'tube' is introduced (tube-MPC, TMPC), forcing the controller to always account for the worst-case state disturbance. This invariant tube can guarantee safety and robustness of the agent [5]. Due to the invariant nature of this tube, the performance of a TMPC agent is significantly worse than standard MPC [4]. As previously mentioned, this is a critical drawback given the time-sensitive nature of SaR. In order to reduce the cost to performance, while maintaining safety and robustness, a state-dependent dynamic TMPC (SDD-TMPC) is proposed, where the controller can vary tube size depending on its state and the environment.

The contribution of this paper lies in exploring how such a controller could be formulated, and how effective such a controller would be. This is explored with a main research question of: how can linking the tube size of a SDD-TMPC agent to said agent's perception of the environment, influence the agent's robustness, conservativeness, and performance in a SaR scenario? In order to answer this question effectively, several sub-questions are proposed:

1. How should disturbance bounds be defined as a function of perceived confidence and disturbance risk, such that performance is improved over TMPC, and empirical robustness is maintained?
2. To what extent are the performance gains and empirical robustness retained under increased perceptual uncertainty?
3. How conservative is the resulting bound compared to a fixed worst-case bound used by conventional TMPC?

The paper aims to provide data and insight regarding these questions, by developing a SaR simulator to act as a testbed for the three controllers: MPC, TMPC, and SDD-TMPC.

The remainder of the paper follows the structure: Section 2, explores work and research related to this paper's topics. Section 3 explains the experiment setup in which the different control models will be tested, and provides a formal definition of the problem. Section 4 details the implementations of the models, includes the resulting data from running the experiment, and analysis of these results. Section 5 discusses the ethical implications of the paper and the research performed. Finally, Section 6 summarizes the findings and interpretations of the paper, and suggests future work and research that can be conducted in this field.

2 Literature Review

The following section explores related work and research regarding the topics of using MPC in SaR, how the performance of TMPC suffers due to being overly conservative, and the gaps of knowledge in this field.

MPC effectively handles input/output constraints, and provides guarantees on stability and feasibility [4]. The MPC algorithm has already been shown to be effective in controlling multiple unmanned aerial vehicles (UAVs) for SaR scenarios [6]. However, due to MPC's aforementioned inability

to compensate for significant state disturbances [4], a tube is introduced as a robust approach.

Despite its robustness, TMPC's conservative assumptions result in low-performance and overly cautious solutions [4]. In respects to SaR, where a quick search time is imperative, this is a critical failure. The suboptimal performance of TMPC and its inferiority to an adaptive tube has been further acknowledged in a multitude of papers, such as the paper on Adaptive Tube MPC by Dey and Bhasin [7].

Recent work explores reducing TMPC's conservativeness through adaptive tube approaches. Dey and Bhasin [7] propose online parameter learning to reduce its conservativeness by gathering data and shrinking the uncertainty parameter set. Similarly, [4] introduces a state-dependent tube, allowing the tube to both shrink and expand based on disturbance levels. Both approaches showed significant improved performance over TMPC. The key difference between these works and the following work, is that this work explores scaling the online dynamic tube on an agent's perception of its environment, and its perceived risk of disturbance.

Despite the vast research which has been done in the field of adaptive-tube MPC, a few knowledge gaps remain. One such gap is the lack of research into SaR-hardware-integrated testing. Many papers experiment with simple motors and single input/output systems [8]. Another gap, which this paper aims to cover, is perception-driven tube adaptation, where the tube-size can vary depending on the environment as perceived by the SaR agent.

3 Methodology

The following section includes a description of the testbed design, used to test the three different agent controllers: MPC, TMPC, and SDD-TMPC, where MPC is included as a non-safety baseline. A formal definition of the problem is then provided. Finally, the section describes the objective function that the controllers optimize for.¹ A table of variable values used for this experiment can be found in Appendix A.

3.1 Testbed

In order to measure the performance of the agent, a 2-dimensional time-varying grid is used to act as a testbed. A set of 5 cell-properties is introduced, those being traversability, vulnerability, fire, victims, and confidence.

Grid

Each property of a cell can be encoded in a $m \times n$ matrix, where the matrices are defined as follows.

A static traversability matrix, $M^{\text{trav}} \in \{0, 1\}^{m \times n}$, where

$$M_{i,j}^{\text{trav}} = \begin{cases} 1, & \text{cell } (i, j) \text{ blocked} \\ 0, & \text{cell } (i, j) \text{ free.} \end{cases} \quad (1)$$

Although traversability = 1 meaning blocked/untraversable is counterintuitive, it is written as such to maintain consistency with the code.

¹Implementation of this methodology can be found at <https://github.com/FilipOBK/SDD-TMPC-Research>.

A static vulnerability matrix, $M^{\text{vuln}} \in [0, 1]^{m \times n}$, where

$$M_{i,j}^{\text{vuln}} \in [0, 1]. \quad (2)$$

A dynamic fire matrix, $M^{\text{fire}} \in \{\text{safe, flammable, burning, burnt}\}^{m \times n}$, where

$$M_{i,j}^{\text{fire}}(k) = \begin{cases} \text{safe,} & \text{cell } (i, j) \text{ is safe at time } k \\ \text{flammable,} & \text{cell } (i, j) \text{ is flammable at time } k \\ \text{burning,} & \text{cell } (i, j) \text{ is burning at time } k \\ \text{burnt,} & \text{cell } (i, j) \text{ is burnt at time } k. \end{cases} \quad (3)$$

A static victim matrix, $M^{\text{vict}} \in \{0, 1\}^{m \times n}$, where

$$M_{i,j}^{\text{vict}} = \begin{cases} 1, & \text{victim present at cell } (i, j) \\ 0, & \text{no victim present at cell } (i, j). \end{cases} \quad (4)$$

Finally, a dynamic confidence matrix, $M^{\text{conf}} \in [0, 1]^{m \times n}$, where

$$M_{i,j}^{\text{conf}}(k) \in [0, 1]. \quad (5)$$

A grid at time step k can formally be defined as the 5-tuple of all of these matrices:

$$G(k) = \langle M^{\text{trav}}, M^{\text{vuln}}, M^{\text{fire}}(k), M^{\text{vict}}, M^{\text{conf}}(k) \rangle \quad (6)$$

The simulation then stores a 'ground truth' grid containing all true values of these matrices, which varies each time step.

As such, at each step the testbed simulator queries every agent's controller about what move it would like to perform next, and is responsible for denying and recording any illegal moves. It is also responsible for recording other statistics and metrics required for analysis of the agents' performance and safety.

Dynamic environment

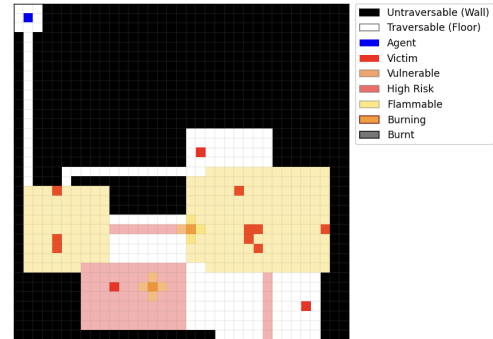


Figure 1: Example of a randomly generated SaR environment, where vulnerable regions represent disturbance probabilities of 0.6, and high-risk regions represent disturbance probabilities of 1.0.

The simulated ground truth is generated dynamically, using the 2D map generation algorithm proposed by [9, Alg. 1]. Seven rooms are generated, and corridors are 1-3 tiles wide. An additional 3x3 starting room is placed in the top left corner, and connected via corridors as described in the algorithm. The agent starts the simulation in the center of this room.

Each room and corridor (let a be the region covered by respective room/corridor) is then iterated through and independently assigned a vulnerability in a two-stage Bernoulli

process. The region is labeled as hazardous with probability p_a . Hazardous regions are then assigned a vulnerability of 1.0 with probability s_a , and a vulnerability of 0.6 otherwise. All cells within the region a inherit the vulnerability of the region. Values 1.0 and 0.6 were chosen as these intervals were found to offer unique challenges for the agents. With $M_{i,j}^{\text{vuln}} = 0.6$, enough scans of the cell would allow for disturbances to be completely mitigated (see Eq. (10)), whereas a $M_{i,j}^{\text{vuln}} = 1.0$ can never be fully mitigated (confidence asymptotically approaches 1.0, see Eq. (11)).

Fire spread is modeled similarly to [10], where a tile burns for duration d , and then becomes burnt. Any tile adjacent to a burning tile is marked as flammable, and a flammable tile has some probability p of igniting at each time-step.

After map generation, victims are placed on 12 random independently selected traversable points on the grid, and 2 random traversable points are selected and marked as burning.

An example of a randomly generated environment can be seen in Fig. 1.

Agents

To allow for informed decision-making and an adaptive tube, agents store a time-varying perception grid $G^{\mathcal{P}}(k)$ with their perceived values of the environment.

At every time step k , an agent 'scans' its local environment, updating $G^{\mathcal{P}}(k)$. Agents further store an 'explored' matrix $E \in \{0, 1\}^{m \times n}$ that stores what cells the agent has scanned at least once:

$$E_{i,j}(k) = \begin{cases} 1 & \text{cell } (i, j) \text{ has been scanned} \\ & \text{at least once at time-step } k, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We can define the agent's state at time-step k as $x_k = \langle (i_k, j_k), G^{\mathcal{P}}(k), E(k) \rangle$, where

$$\begin{aligned} (i_k, j_k) &\in \mathcal{S} = \{(i, j) \mid i \in [1..n], j \in [1..m]\}, \\ G^{\mathcal{P}}(k) &\in \mathbb{P}, \\ E(k) &\in \{0, 1\}^{n \times m} \end{aligned} \quad (8)$$

where $\mathbb{P}$ is the set all possible perceived grids.

A cell is scanned if the euclidean distance to the agent is less than the scan radius r , as well as if the agent has line of sight to the cell. Line of sight is calculated using the python library `tcod`, specifically, the `tcod.map.compute.fov` method [11]. An agent also has attributes 'scan accuracy' η_0 , and 'confidence decay' σ . To simulate increasing uncertainty when scanning from further away (due to smoke or other environmental factors), the increase in confidence when scanning a cell (i, j) from position (i_k, j_k) is defined similar to a normal distribution centered on the cell the agent occupies:

$$\eta_{i_k, j_k}(i, j) = \eta_0 \exp \left(-\frac{(i - i_k)^2 + (j - j_k)^2}{2 \left(\frac{r}{3}\right)^2} \right). \quad (9)$$

$$M_{i,j}^{\text{conf}}(k+1) = \begin{cases} M_{i,j}^{\text{conf}}(k) + \Delta_{i,j}(k+1), & \text{scanned,} \\ \max(0, M_{i,j}^{\text{conf}}(k) - \sigma), & \text{not scanned,} \end{cases} \quad (10)$$

where the confidence increment for a scanned cell is given by:

$$\Delta_{i,j}(k) = \eta_{i_k, j_k}(i, j) \cdot (1 - M_{i,j}^{\text{conf}}(k-1)). \quad (11)$$

In order to gather data for research sub-question 2, an alternative scanning procedure for the vulnerability matrix is introduced. In a normal scan, the true values of scanned cells S_k are read directly from the ground truth grid $G(k)$ matrices:

$$M_{i,j}^{*,\mathcal{P}}(k) = M_{i,j}^*(k), \quad \forall (i, j) \in S_k. \quad (12)$$

The alternative vulnerability scan introduces uniform noise of magnitudes $\delta \in \{0.1, 0.2, 0.4\}$, representing increasing levels of uncertainty in vulnerability readings:

$$\begin{aligned} M_{i,j}^{\text{vuln},\mathcal{P}}(k) &= \min(1, \max(0, M_{i,j}^{\text{vuln}} + \epsilon_{i,j}(k))), \\ \epsilon_{i,j}(k) &\sim U(-\delta, \delta), \quad \delta \in \{0.1, 0.2, 0.4\}. \end{aligned} \quad (13)$$

The value 0.1 was chosen to introduce small noise around each of the vulnerability levels. 0.2 was chosen as it spans a total range of 0.4, matching the gap between the two ground truth vulnerabilities (0.6, 1.0). Finally, 0.4 was chosen as it significantly exceeds the gap, making it possible for the noise to overshadow the ground truth vulnerability. This range of noise magnitudes and effects aims to offer an interesting variety of behavior for analysis.

3.2 Problem Definition

The search and rescue scenario is modeled as a 2-dimensional discrete grid of width n and height m , where changes and actions occur during discrete time-steps. The following problem is based heavily on the state-dependent dynamic tube MPC proposed by [4]. As such, the following formalization of the problem will be closely related.

Sets

We introduce a state vector x_k , representing an agent's location, and a control input u_k . This state vector can be affected by additive state-dependent external disturbances $w(x_k)$ at time-step $k \in \mathbb{N}$, where $x_k \in \mathbb{X} \subseteq \mathbb{N}^n$, $u_k \in \mathbb{U} \subseteq \mathbb{Z}^2$, and $w(x_k) \in \mathbb{W} \subseteq \mathbb{Z}^2$. The admissible set $\mathbb{X}$, and the control input set $\mathbb{U}$ are static, while the external disturbance set $\mathbb{W}$ can vary as a function of the state x_k . The control input set is defined as follows:

$$\mathbb{U} = \left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \quad (14)$$

representing the actions UP, DOWN, LEFT, RIGHT, and WAIT respectively.

We further define the feasible set $\mathcal{X}$, the set of all states that satisfy the hard constraints. States break hard constraints if the agent is occupying a cell that: is burning, has a victim, is untraversable, or is out of bounds of the grid. Formally:

$$\begin{aligned} \mathcal{X} &= \{x_k \in S \times \mathbb{P} \times \{0, 1\}^{m \times n} \mid (i_k, j_k) \in S, \\ M_{i_k, j_k}^{\text{trav}, \mathcal{P}} &= 0, M_{i_k, j_k}^{\text{fire}, \mathcal{P}}(k) \neq \text{burning}, M_{i_k, j_k}^{\text{vict}, \mathcal{P}} = 0\}. \end{aligned} \quad (15)$$

The external disturbance set is defined as:

$$\mathbb{W} = \mathbb{U} \setminus \{(0, 0)^{\top}\}, \quad (16)$$

where the vector $(0, 0)^{\top}$ is omitted since this would represent a trivial case of no disturbance.

Let $v(x_k)$ denote the probability of a disturbance occurring at cell (i, j) occupied by the SaR-agent at time-step k :

$$v(x_k) = \max \left(0, M_{i,j}^{\text{vul}} - M_{i,j}^{\text{conf}}(k) \right)^2. \quad (17)$$

$v(x_k)$ is defined such that when the agent slows down and scans a cell multiple times, its confidence on that cell increases, decreasing the likelihood that a disturbance occurs. The confidence at a cell can only ever approach 1, never reach it (see Eq. (10)), meaning disturbances in a high-risk area can never be fully mitigated. An actual instance of disturbance is modeled as a random variable w_k taking values in $\mathbb{W}$ according to:

$$\Pr(w_k = \omega \mid x_k) = \begin{cases} \frac{v(x_k)}{|\mathbb{W}|}, & \omega \in \mathbb{W}, \\ 1 - v(x_k), & \omega = (0, 0)^\top. \end{cases} \quad (18)$$

That is, at any time-step k , w_k is a disturbance with probability $v(x_k)$, and w_k is no disturbance with probability $1 - v(x_k)$.

The dynamic system can now be defined similarly to [4, Eq. (1)]:

$$x_{k+1} = f(x_k, u_k, w_k), \quad (19)$$

where $f : \mathbb{X} \times \mathbb{U} \times \mathbb{W} \rightarrow \mathbb{X}$ is the state transition function. The returned state is at the new position calculated by the move and disturbance, and has an updated perception grid $G^P(k+1)$. An agent's predicted next state $x_{k+1|k}$ applies the same state transition function, but predicts the fire-spread rate to be $p = 1.0$.

The goal of this dynamic system is to maximize an objective function J over a prediction horizon N , while all hard constraints are kept satisfied; for any time-step k . This is just as in [4, Eq. (2)], but maximizing, rather than minimizing:

$$\begin{aligned} J^*(x_k) &= \max_{\tilde{x}_k, \tilde{u}_k} J(\tilde{x}_k, \tilde{u}_k) \\ \text{s.t. for } i &= k, \dots, k+N-1, \\ x_{i+1|k} &= f(x_{i|k}, u_{i|k}, w(x_{i|k})), \\ x_{k|k} &= x_k, \\ x_{i+1|k} &\in \mathbb{X}, \\ u_{i|k} &\in \mathbb{U}, \\ w(x_{i|k}) &\in \mathbb{W}, \end{aligned} \quad (20)$$

where $\tilde{x}_k = (x_{k+1|k}, \dots, x_{k+N|k})$, where $x_{i|k}, i > k$ is the predicted state of the agent at time-step i , given the agent at time-step k . Similarly, $\tilde{u}_k = (u_{k|k}, \dots, u_{k+N-1|k}), i \geq k$, where the value of u_i is determined at time-step k , and J^* is the optimal value of the objective function found over an exhaustive search of $\tilde{x}_k, \tilde{u}_k$. The objective function J is composed of stage objectives accumulated over the horizon N , as well as a terminal objective calculated at the terminal time-step $k+N$. The objective function is continuous and finite for $x_i \in \mathbb{X}, \forall i \in \{k, \dots, k+N-1\}$.

As noted by [12], because fire is a dynamic obstacle, and the environment is partially observable, it is impossible for the agent to guarantee the feasibility at the beginning of the scenario, and as such recursive feasibility is also impossible. It is, therefore, not explored in this research.

Finally, a successful run is defined as the agent having scanned all 12 cells that contain a victim, at least once, without ever having broken a constraint. Breaking a hard constraint is treated as a failure, as any collision with a wall could lead to damage of the robot, while collisions with a victim could be detrimental to the SaR. Formally, the run is a success at time k if:

$$\sum_{i=1}^m \sum_{j=1}^n E_{i,j}(k) \cdot M_{i,j}^{\text{vict}} = 12 \quad \wedge \quad x_i \in \mathcal{X}, \forall i \in \{1, \dots, k\} \quad (21)$$

As further stated in [12], dynamic fire and partial observability also makes constraint satisfaction at every time-step impossible; it is not always possible for the controller to avoid getting trapped into a corner by fire without a feasible move to escape, resulting in fire collisions. Due to the lack of agency the controllers have over fire collisions, these are treated separately from controlled-induced constraint violations. This is further discussed in Sec. 4.2.

3.3 Objective Function

The following subsection describes the main objective function that all three controllers optimize for.

The objective function is split into stage objectives accumulated over the agents' exhaustive search of moves over horizon N , and a terminal objective, evaluated on the state at the horizon. At time step k , the stage objective at predicted time-step $i \in [k+1..k+N]$ is described as follows:

$$\begin{aligned} J_{\text{stage},k}(x_{i|k}) &= \frac{200}{\pi r^2} \left(\sum_{i=1}^m \sum_{j=1}^n E_{i,j}(i) - E_{i,j}(i-1) \right) \\ &\quad + \frac{3}{m \times n} \left(\sum_{i=1}^m \sum_{j=1}^n M_{i,j}^{\text{conf}}(i) \right) \end{aligned} \quad (22)$$

(see Eq. (7) for definition of E). Note that the objective solely rewards exploration and confidence. Because safety is handled completely through admissible-state constraints, the controller does not consider trade-offs between safety and the objective.

The terminal objective, at time-step $k+N$, uses a breadth-first-search function executed at time k to calculate the distances from every cell to an unexplored cell. This function D_k is defined as:

$$\begin{aligned} \mathcal{F}_k &= \{(i, j) \mid E_{i,j} = 0\} \\ D_k(i, j) &= \min_{(u,v) \in \mathcal{F}_k} d_{sp}((i, j), (u, v)) \end{aligned} \quad (23)$$

where d_{sp} denotes the shortest-path distance along cells in the admissible set $\mathbb{X}$. The terminal objective is then defined as:

$$J_{\text{terminal},k}(i, j) = -\frac{300 \cdot N \cdot D_k(i, j)}{m + n}, \quad (24)$$

resulting in the final objective function:

$$\begin{aligned} J^*(x_k, u_k) &= \max_{\tilde{x}_k, \tilde{u}_k} \sum_{i=k}^{k+N-1} J_{\text{stage},k}(x_{i|k}) \\ &\quad + J_{\text{terminal},k}(i_{k+N|k}, j_{k+N|k}) \end{aligned} \quad (25)$$

4 Results

The following section describes the implementation details of the three controller models. As mentioned in Sec.3, MPC is included for completeness, and as a non-safety baseline. This section further contains the processed data from performing 250 differently seeded generated simulations. Three additional SDD-TMPC controllers, using the alternative scanning approach described in Eq. (13) with the different delta values, are also recorded. In total, six sets of 250 seeded runs are performed. From previous testing it was determined that if a scenario was feasible for a controller, it would find all 12 victims within 300 time-steps. Therefore, if an agent does not find all victims within 300 time-steps, the run is labeled as a failure and halted. The section further contains analysis of the data, and its connections to the research questions.

4.1 Model Implementations

All three models solely optimize for the objective function (Eq. (25)) as described in Eq. (20). The differences between the models are solely the sets of admissible states $\mathbb{X}$ for each controller.

MPC

The (nominal) MPC agent was implemented without any consideration of vulnerability or disturbances. As such, the MPC's admissible set $\mathbb{X} = \mathcal{X}$.

TMPC

TMPC 'expands the tube' by expecting the maximal disturbance for any move. The admissible set can as such be defined using the Minkowski sum (denoted as $\oplus$), and the feasible set $\mathcal{X}$:

$$\mathbb{X} = \{x \mid x \in \mathcal{X} \text{ and } \omega \in \mathcal{X}, \forall \omega \in \{x\} \oplus \mathbb{W}\} \quad (26)$$

SDD-TMPC

As SDD-TMPC has a dynamic tube, the admissible set is dynamic per time-step k . The admissible set becomes state-dependent:

$$\mathbb{X}(x_k) = \begin{cases} \mathcal{X}, & v(x_k) = 0, \\ \mathcal{T}, & v(x_k) > 0, \end{cases} \quad (27)$$

$$\mathcal{T} = \{x \mid x \in \mathcal{X} \text{ and } \omega \in \mathcal{X}, \forall \omega \in \{x\} \oplus \mathbb{W}\}.$$

That is, when the perceived disturbance probability (Eq. (17)) is 0, SDD-TMPC reduces to nominal MPC. Otherwise, states reachable in one disturbance step must also be in the feasible set (reduces to TMPC). This aims to be safe by ensuring that a reduction at any point still enforces constraint satisfaction over all states reachable from the current. An interesting effect of the discrete grid model is that the SDD-TMPC formulation becomes binary, where its tube reduces to one of the other two controller tubes. While this is a direct consequence of the discrete model, it does mean that continuously varying tube sizes is not explored.

Note that when performing exhaustive search, at each iteration, the admissible set is also predicted ($\mathbb{X}(x_{i|k})$). Additionally, when performing the frontier-distance BFS an artificial confidence matrix $M^{\text{conf}} = \{0.999\}^{m \times n}$ is used, meaning that the SDD-TMPC exploration heuristic (see Eq. (23)) is

calculated using the largest possible admissible set. The artificial confidence matrix ensures that the agent is not restricted to only areas reachable given its current confidence, rather it can plan towards areas that can become accessible with further scanning. A value of 1.0 is avoided since confidence can only asymptotically approach 1.0 (Eq. (10)), and for cells where $M^{\text{vuln}} = 1.0, v(x_k) > 0$. Using artificial confidence = 1.0 would incorrectly allow the agent to plan routes through corridors it can never safely traverse. Importantly, trajectory feasibility is still checked using the actual confidence, to determine disturbance probability and maintain safety.

4.2 Data and Analysis

The following subsection aims to analyze the data gathered over the six sets of 250 seeded runs. Every subsection aims to connect the implications of the data to its respective research sub-questions listed at the end of Sec. 1.

Perception to Tube Size Mapping

The following subsection aims to explore if the mapping proposed in Eq. (27) is effective at maintaining the safety of TMPC, while improving performance.

Controller	Collisions		
	Walls	Victims	Fires
MPC	206	44	7
TMPC	0	0	42
SDD-TMPC	0	0	21

Table 1: Total accumulated collision of each type per controller, over 250 runs. TMPC and SDD-TMPC have 0 wall and victim collisions. Notably, they have fire collisions, from being backed into corners by fire.

Table 1 depicts the total collisions per controller over the 250 runs. Importantly, when implementing the robust tubes proposed in Sec. 4.1, the collisions with the static fields, walls and victims, drops to 0.

Notably, collisions with fire remain. Manually observing the data, these collisions occur rarely, in only 3 of the 250 runs. These stem from the agent being cornered by spreading fire with no remaining feasible move (as described in Sec. 3.2), rather than from a failure of the disturbance tube to bound the agent's reachable states. The data is included for completeness, but because the goal of the research is to evaluate controller correctness, the collisions are excluded when comparing safety and successes between the controllers. This exclusion is applied throughout all controllers and delta values.

Since wall and victim collisions are entirely absent for both TMPC and SDD-TMPC, there are no discordant pairs between the two. That is, for every seed that is collision free under TMPC, it is also collision free under SDD-TMPC and vice versa. As such, a McNemar test [13] is not informative here. For the precision of this observation of zero wall/victim collisions, a 95% Wilson confidence interval is used, corresponding to a CI of [0%,1.5%].² This suggests that de-

²Wilson score interval, n=250, k=0.

spite the finite test set, the actual collision rates of these controllers are unlikely to exceed 1.5%. In regards to research sub-question 1, this clearly indicates that the empirical robustness of TMPC is maintained in SDD-TMPC.

The second part of sub-question 1 addresses an improvement in performance over TMPC. One measure of performance would be the rate of success, as defined at the end of Sec. 3.2.

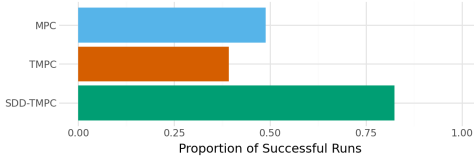


Figure 2: Success rates by controller: proportion of runs that find all victims within 300 time-steps without breaking hard constraints (ignoring fire collisions). Rates: MPC: 0.488, TMPC: 0.398, SDD-TMPC: 0.824.

Figure 2 displays expected results, the baseline pure nominal MPC’s struggles to succeed due to its lack of safety. Interestingly, MPC’s rate of success still exceeds that of the overly conservative TMPC. However, this can be explained by: when in a situation that might result in a collision, TMPC will never overcome it (it aims to guarantee safety), whereas MPC will attempt to overcome it, with a not-insignificant probability of succeeding. In order to determine the correlation between TMPC and SDD-TMPC’s success rates, an exact pairwise McNemar test is performed, resulting in a statistic of 1.0 and $p = 1.71 \times 10^{-31}$. With $p < 0.001$, the null-hypothesis that utilizing the proposed mapping does *not* have a significant statistical effect on success rate can be rejected. To understand the extent of the benefit to performance SDD-TMPC offers over TMPC, the odds-ratio (OR) is calculated. Of the 250 runs 108 succeeded under only SDD-TMPC, whereas only 1 succeeded under only TMPC (OR= 40.0, 95% CI [5.41, 295.75]).

The second aspect of performance is the rate at which the controller is able to locate the 12 victims. Evaluation of mission durations is done only with seeds where both controllers succeeded, as failed runs correspond to a fixed timeout of 300 time-steps and do not represent true completion times. Including these would conflate performance with success rate. These differences are, however, accounted for in the above success-rate analysis. Among the selected seeds, there was a statistically significant difference in mission duration between TMPC and SDD-TMPC (Wilcoxon signed-rank test, $W = 1612.0, p = 0.0059$ [14]). The median duration decreased from 170.0 steps (TMPC) to 153.0 (SDD-TMPC), a median-paired improvement of 17 steps.

Overall, the data statistically supports that the empirical robustness of TMPC is maintained by the mapping proposed in Eq. (27), with no wall/victim collisions observed over the 250 runs. The mapping further provides a statistically significant performance improvement, that is, SDD-TMPC has a higher rate of success, and in scenarios where both succeed, SDD-TMPC a lower mission duration (median).

Effects of Uncertainty

The following subsection aims to explore how well robustness and performance is maintained under increasing perceptual uncertainty. This uses the data collected from performing 3 sets of 250 SaR simulations, with the different noise/delta values listed in Eq. (13).

Delta	Collisions		
	Walls	Victims	Fires
0.0	0	0	21
0.1	0	0	7
0.2	0	0	7
0.4	5	4	39

Table 2: Total accumulated collision of each type per delta value, over 250 runs. 0 wall- and victim collisions are maintained from $\delta = 0$ to $\delta \in \{0.1, 0.2\}$, but not at $\delta = 0.4$.

As shown in Tab. 2, for delta values 0.1, and 0.2, the static wall and victim collision remain unaffected when compared to $\delta = 0$. The fire collisions are once again included for completeness, but due to the previously mentioned reasons, cannot be used as a fair comparison for the safety and success between delta values. With 250 runs for each delta, the 95% Wilson CI for zero observed collisions remains the same at [0%, 1.5%]. For $\delta = 0.4$, 9 runs are observed with wall/victim collisions, an empirical collision rate of 3.6% (95% CI [1.9%, 6.7%]).

Performing an *exact* pairwise McNemar test on collisions between the runs with delta values 0.2 and 0.4, results in a statistic of 0.0, and a p-value of 3.91×10^{-3} . At the statistical significance level of $\alpha = 0.05$, the null-hypothesis that the collision rate is not significantly impacted by the increase in delta, can be rejected. All 9 discordant pairs correspond to a collision under $\delta = 0.4$, where no collision occurred under $\delta = 0.2$, confirming that $\delta = 0.4$ represents a statistically significant loss of empirical safety maintained at $\delta \leq 0.2$. This clearly demonstrates to which extent empirical robustness is maintained under increasing environmental uncertainty.

The second research sub-question also addresses to which extent performance is retained under the increasing environmental uncertainty.

Delta	# Successful Runs	Success Rate
0.1	212	0.848
0.2	197	0.788
0.4	188	0.752

Table 3: Number of successful runs out of 250 runs, and respective rates of success of SDD-TMPC across delta values 0.1, 0.2, 0.4.

To determine whether there is a statistically significant difference in success-rate between deltas, a Cochran’s Q test [15] is performed, the results of which are $Q = 11.6, p = 3.02 \times 10^{-3}$. The test does indicate that there is a significant statistical difference, where Tab. 3 displays how the success rate decreases monotonically from $\delta = 0.1$ to $\delta =$

0.4. Interestingly, the success rate when $\delta = 0.1$ (0.848, Tab. 3), is marginally higher than when $\delta = 0.0$ (0.824, Fig. 2). This can be explained similarly to why the success rate of MPC slightly exceeds that of TMPC, where the small amount of noise relaxes how conservative the SDD-TMPC agent is, making previously infeasible regions appear traversable. Since the perceived vulnerability is still high, the controller is still extremely cautious when traversing these regions. However, for $\delta \geq 0.2$, the increased mismatch in perceived and ground truth vulnerability outweighs the benefit, causing success-rates to decline.

Delta	Success OR	95% CI
0.1	2.33	[1.05, 5.17]
0.2	3.02	[1.46, 6.21]
0.4	1.35	[0.74, 2.47]

Table 4: Success ORs of SDD-TMPC at different delta values, compared to TMPC successes.

Table 4 presents the odds ratios of SDD-TMPC success against TMPC across noise levels. At $\delta \in \{0.1, 0.2\}$, SDD-TMPC retains its statistically significant performance advantage over TMPC. At $\delta = 0.4$, however, the OR drops to 1.35 with a CI crossing 1.0, indicating that the performance advantage is no longer statistically significant. This identifies $\delta = 0.4$ as the boundary point, where increased perceived noise outweighs the advantage of SDD-TMPC over TMPC, which is consistent with the safety degradation seen at the same delta in Tab. 2.

Taking all of this into consideration, SDD-TMPC statistically retains its robustness and performance improvement over TMPC for $\delta \leq 0.2$. At $\delta = 0.4$, safety begins to degrade with observed wall/victim collisions, and the success OR of SDD-TMPC over TMPC becomes statistically insignificant. It is clear that there is a boundary of perceptual noise in $\delta \in (0.2, 0.4]$, where empirical robustness and performance advantage are no longer retained.

Conservativeness Comparison

The following subsection explores how conservative the resulting bound of the SDD-TMPC formulation is compared to the fixed-worst case bound used by conventional TMPC.

As the maximum disturbance magnitude in this model is an offset of 1 tile, the worst-case bound used by TMPC is designated a tube size of 1. Similarly, as MPC *never* considers disturbances (always assumes a disturbance of magnitude 0), the MPC ‘tube’ is designated a size of 0. This definition allows for the conservatism of SDD-TMPC to be quantified through its tube size, and compared against TMPC. Note that due to the binary nature of the tube, the average tube size over a run becomes equivalent to the fraction of time the SDD-TMPC agent spends in ‘TMPC mode’.

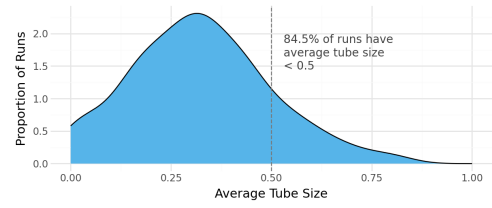


Figure 3: Density of average SDD-TMPC tube sizes over 206 successful runs / 250 runs. 84.5% of runs have an average tube size < 0.5 .

Over the 250 runs performed, at each run, the average tube size of the SDD-TMPC agent was recorded. As stated in Fig. 3, in 84.5% of the runs, the average tube size of the agent was < 0.5 , making it clear that under the tested conditions, SDD-TMPC was on average substantially less conservative than TMPC. For a more concrete metric, the mean of SDD-TMPC’s average tube sizes was computed. The calculated mean was 0.323 (95% bootstrap CI³ [0.299, 0.347]), indicating that SDD-TMPC operated in the tightened TMPC-like mode for only 32.3% of successful run time-steps on average, which represents a significant decrease in conservativeness.

Taking all of this into consideration, it is clear that the bound of SDD-TMPC is significantly less conservative than that used by the fixed worst-case TMPC, while maintaining empirical safety as explored in the first research sub-question.

5 Responsible Research

The following section discusses the ethical aspects and implications of this research, as well as the reproducibility of the methods used to conduct it.

5.1 Ethics

The context of the research places the agent models in a SaR environment and scenario, as such an environment allows for complex interactions and challenges that allow for detailed evaluation of the controllers’ behavior. However, it is important to acknowledge that many details of a real-world SaR scenario were not modeled. In the case that this research is to be implemented in a real-world SaR scenario, great care should be taken to consider discrepancies between the modeled and real-world scenarios, in order to ensure people’s safety.

5.2 Use of Generative AI

The Anthropic generative AI *Claude* was used sparingly, and following the guidelines set by TUDelft. Use was limited to: LaTeX formatting assistance, grammar and spell checking, and literature search support.

5.3 Reproducibility

Throughout the development of the testbed, and the agent controllers, great care was spent to ensure that only open-source, and freely available software were used. This was done in an attempt to make the research as reproducible as

³Bootstrap CI used as tube-size distribution is non-normal.

possible. Moreover, all code and raw data that was used to create and evaluate the models, was made open-source to further allow for reproducibility.

6 Conclusion

The following section summarizes the goals and findings of the research. Furthermore, it highlights interesting and noteworthy aspects of the research, and offers ideas of future work and experiments that can be done to further understanding of this area.

6.1 Summary

The main goal of this research was to explore how a dynamic tube can be constructed, such that it is linked to an agent’s perception of the environment, and how that would influence said agent’s safety, and performance. From this, three sub-questions were derived. First, how can an agent’s perceived risk and confidence be mapped to a tube size, such that the agent’s performance is improved over TMPC, while empirical robustness is maintained? Second, how does changing the uncertainty of the measurements by the agent affect its robustness and performance. Third, how conservative is such a tube, when compared to a fixed worst-case tube used by conventional TMPC?

In regards to the first research sub-question, a mapping that scaled tube size on the difference between perceived risk and perceived confidence was developed. This mapping was observed to effectively maintain the empirical safety of fixed worst-case TMPC, with no wall nor victim collisions observed over 250 runs. A statistically significant advantage in performance of SDD-TMPC over TMPC was found, with an OR on successful runs of 40.0 (95% CI [5.41, 295.75]), and a median-paired improvement of 17 time-steps on mission duration. A noteworthy limitation of the model, is that because the agent does not have perfect information, the controller cannot always account for dynamic obstacles such as fire, resulting in a small number of fire collisions observed. These are not failures of the disturbance tube, rather an inherent limitation of the model and any controller under it. However, this inability to account for dynamic hazards without perfect information does bring into question how applicable these techniques are for real-world search and rescue. Methods to address such dynamic hazards given partial information should be investigated further.

For the second research sub-question, introducing varying levels of noise into the SDD-TMPC agent’s vulnerability readings was found to have statistically significant effects on the controller’s safety and performance. Uniformly sampled noise of magnitudes $\delta \in \{0.1, 0.2, 0.4\}$ were tested, where a statistically significant performance advantage over TMPC was retained up to $\delta = 0.2$ (OR=3.02, 95% CI [1.46, 6.21]). $\delta = 0.4$ was identified as the boundary where both robustness and the performance advantage were no longer retained by SDD-TMPC over TMPC. Overall, the SDD-TMPC formulation retained empirical safety and performance gains to the extent that the magnitude of the noise did not fully overshadow the ground-truth values.

Finally, in regards to the third research sub-question, the dynamic tube scaling approach of SDD-TMPC was found to

be drastically less conservative than worst-case TMPC. Due to the discrete model designed for this research, the tube becomes binary, either reducing to MPC when there is no perceived risk of a disturbance, or reducing to TMPC when there is a perceived risk of a disturbance. The data collected showed that SDD-TMPC operated in ‘TMPC mode’ for only 32.3% (95% bootstrap CI [0.299, 0.347]) of time-steps in successful runs, representing a significant reduction in conservativeness compared to fixed worst-case TMPC. As noted in Sec. 4.1, while the binary model is an interesting detail, it is also a limitation of the current method, as it does not yet test continuously varying tube sizes.

Overall, linking the tube size of a SDD-TMPC agent to its perception of the environment, is a viable and effective approach. The controller is able to maintain the empirical robustness of fixed worst-case TMPC, while substantially increasing success rate and search rate. Furthermore, the SDD-TMPC was found to be significantly less conservative than TMPC. These benefits were found to be retained under significant perceptual noise, only being lost when the magnitude of the noise was so large that it overshadowed the true readings.

6.2 Future Work

The SDD-TMPC mapping design and empirical robustness are subject to the fact that the agent receives perfect information when regarding the type of cell it scans (i.e. an agent will never confuse a victim for a wall). This is unrealistic, a limitation of the model proposed for this research. As such, an interesting possibility for further research would be to introduce probabilities into the different perception matrices of the agent (e.g. after scanning, cell (i, j) has a 30% chance of being a wall, and 70% chance of being a victim). A more accurate/realistic model could provide a deeper insight into possible real-world challenges that agents might encounter when performing critical tasks such as search and rescue.

Another limitation of this work, which could be addressed in future research, is the discrete nature of the model; that is, agents move 1 cell every discrete time step, and only in cardinal directions. Similarly, disturbances only offset the agent’s position by 1 cell in a cardinal direction. A goal of future research could be to develop a perception-based tube scaling function for a continuous linear system. The magnitude of a disturbance would not be binary such as in this case, rather exist in a range, similar to non-perception based research such as [4, 6].

A third limitation of this work is the relatively simple model of search and rescue. In this work, victims remain static/stationary and unresponsive. Although this may be true for some victims, it is also likely for there to be panic and dynamic victims in such an environment. Future work could introduce dynamic victims, as well as functionality allowing the agent to aid victims, especially when near hazards such as vulnerable areas or fire, similar to [16]. This could improve the feasibility of applying these concepts to real-world situations, reducing the risk to human life in SaR scenarios.

A Experiment variable values

Variable	Description	Value
m	grid height	35
n	grid width	35
r	scan radius (eq. (9))	4
η_0	scan accuracy (eq. (9))	0.35
σ	confidence decay (eq. (9))	0.05
p_a	hazard probability	0.3
s_a	severity probability	0.4
d	fire burn duration	14
p	fire spread probability	0.04
N	prediction horizon	4
p_u	unconnected probability	0
w_{rmin}	minimum room width	6
w_{rmax}	maximum room width	12
h_{rmin}	minimum room height	6
h_{rmax}	maximum room height	12

Table 5: Experiment setup variable values

References

- [1] J. Liu, Y. Wang, B. Li, and S. Ma, “Current research, key performances and future development of search and rescue robots,” *Frontiers of Mechanical Engineering in China*, vol. 2, no. 4, pp. 404–416, 2007, ISSN: 1673-3592. DOI: 10.1007/s11465-007-0070-2. [Online]. Available: <https://doi.org/10.1007/s11465-007-0070-2>.
- [2] J. G. Blitch, “Artificial intelligence technologies for robot assisted urban search and rescue,” *Expert Systems with Applications*, vol. 11, no. 2, pp. 109–124, 1996, Army Applications of Artificial Intelligence, ISSN: 0957-4174. DOI: [https://doi.org/10.1016/0957-4174\(96\)00038-3](https://doi.org/10.1016/0957-4174(96)00038-3). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0957417496000383>.
- [3] A. L. Adams et al., “Search is a time-critical event: When search and rescue missions may become futile,” *en, Wilderness Environ Med*, vol. 18, no. 2, pp. 95–101, 2007.
- [4] F. Surma and A. Jamshidnejad, “State-dependent dynamic tube mpc: A novel tube mpc method with a fuzzy model of disturbances,” *International Journal of Robust and Nonlinear Control*, vol. 35, no. 4, pp. 1319–1354, 2025. DOI: <https://doi.org/10.1002/rnc.7558>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.7558>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.7558>.
- [5] W. Langson, I. Chrysoschoos, S. Raković, and D. Mayne, “Robust model predictive control using tubes,” *Automatica*, vol. 40, no. 1, pp. 125–133, 2004, ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2003.08.009>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109803002838>.
- [6] F. A. de Alcantara Andrade et al., “Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control,” *Sensors*, vol. 19, no. 19, 2019, ISSN: 1424-8220. DOI: 10.3390/s19194067. [Online]. Available: <https://www.mdpi.com/1424-8220/19/19/4067>.
- [7] A. Dey and S. Bhasin, *Adaptive tube mpc: Beyond a common quadratically stabilizing feedback gain*, 2026. arXiv: 2603.15912 [eess.SY]. [Online]. Available: <https://arxiv.org/abs/2603.15912>.
- [8] S. Jafari Fesharaki, M. Kamali, and F. Sheikholeslam, “Adaptive tube-based model predictive control for linear systems with parametric uncertainty,” *IET Control Theory & Applications*, vol. 11, no. 17, pp. 2947–2953, 2017. DOI: <https://doi.org/10.1049/iet-cta.2017.0228>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-cta.2017.0228>. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cta.2017.0228>.
- [9] G. O. Flores-Aquino, J. Duvier Díaz Ortega, R. Y. Almazan Arvizu, O. Octavio Gutierrez-Frias, R. L. Muñoz, and J. Irving Vasquez-Gomez, “2d grid map generation for deep-learning-based navigation approaches,” in *2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, 2021, pp. 66–70. DOI: 10.1109/ICMEAE55138.2021.00018.
- [10] P. Bak, K. Chen, and C. Tang, “A forest-fire model and some thoughts on turbulence,” *Physics Letters A*, vol. 147, no. 5, pp. 297–300, 1990, ISSN: 0375-9601. DOI: [https://doi.org/10.1016/0375-9601\(90\)90451-S](https://doi.org/10.1016/0375-9601(90)90451-S). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/037596019090451S>.
- [11] B. Kyle. “Python-tcod 21.2.0 documentation.” *tcod v21.2.0 Documentation*. [Online]. Available: <https://python-tcod.readthedocs.io/en/latest/index.html>.
- [12] S. Yan, P. Goulart, and M. Cannon, “Stochastic model predictive control with discounted probabilistic constraints,” in *2018 European Control Conference (ECC)*, IEEE, 2018, pp. 1003–1008. DOI: 10.23919/ecc.2018.8550520. [Online]. Available: <http://dx.doi.org/10.23919/ECC.2018.8550520>.
- [13] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947. DOI: 10.1007/BF02295996.
- [14] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.
- [15] W. G. Cochran, “The comparison of percentages in matched samples,” *Biometrika*, vol. 37, no. 3/4, pp. 256–266, 1950, ISSN: 00063444, 14643510. Accessed: Jun. 18, 2026. [Online]. Available: <http://www.jstor.org/stable/2332378>.

- [16] M. Grouev, “Human-robot interaction in search and rescue: Modelling victim behaviour and trust,” Bachelor’s Thesis, Delft University of Technology, 2026.