Transportation Research Part C 154 (2023) 104268

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Multiple-Input-Single-Output prediction models of crowd dynamics for Model Predictive Control (MPC) of crowd evacuations ☆

Miguel A. Lopez-Carmona *, Alvaro Paricio Garcia

Universidad de Alcala, Escuela Politecnica Superior, Departamento de Automatica, Campus Externo de la Universidad de Alcala, Alcala de Henares, Madrid, Spain

ARTICLE INFO

Keywords:
Controlled evacuation
Pedestrian flow
Crowd dynamics
System identification
State-space model
Input–output model
Prediction model

ABSTRACT

Predicting crowd dynamics in real-time may allow the design of adaptive pedestrian flow control mechanisms that prioritize attendees' safety and overall experience. Single-Input-Single-Output (SISO) AutoRegresive eXogenous (ARX) prediction models of crowd dynamics have been effectively used in Linear Model Predictive Controllers (MPC) that adaptively regulate the movement of people to avoid overcrowding. However, an open research question is whether Multiple-Input, State-space, and Nonlinear modeling approaches may improve MPC control performance through better prediction capabilities. This paper considers a simulated controlled evacuation scenario, where evacuees in a long corridor dynamically receive speed instructions to modulate congestion at the exits. We aim to investigate Multiple-Input-Single-Output (MISO) prediction models such that the inputs are the control action (speed recommendation) and pedestrian flow measurement, and the output is the local density of the pedestrian outflow. State-space and Input–output MISO models, linear and neural, are identified using a data-driven approach in which input–output datasets are generated from strategically designed microscopic evacuation simulations. Different estimation algorithms, including the subspace method, prediction error minimization, and regularized AutoRegressive eXogenous (ARX) model reduction, are evaluated and compared. Finally, to investigate the importance of measuring and modeling the pedestrian inflow, the case in which the models' structure is defined as a Single-Input-Single-Output (SISO) system has been explored, where the pedestrian inflow is considered an unmeasured input disturbance. This study has important implications for the design of more effective MPC controllers for regulating pedestrian flows. We found that the prediction error minimization algorithm performs best and that nonlinear state-space modeling does not improve prediction performance. The study suggests that modeling the inner state of the evacuation process through a state-space model positively influences predicting system dynamics. Also, modeling pedestrian inflow improves prediction performance from a predefined prediction horizon value. Overall, linear state-space models have been deemed the most suitable option in corridor-type scenarios.

☆ This article belongs to the Virtual Special Issue on "Pedestrians & Crowds".

* Corresponding author.
E-mail addresses: miguelangel.lopez@uah.es (M.A. Lopez-Carmona), alvaro.paricio@uah.es (A. Paricio Garcia).

https://doi.org/10.1016/j.trc.2023.104268
Received 20 April 2023; Received in revised form 19 July 2023; Accepted 22 July 2023
Available online 31 July 2023

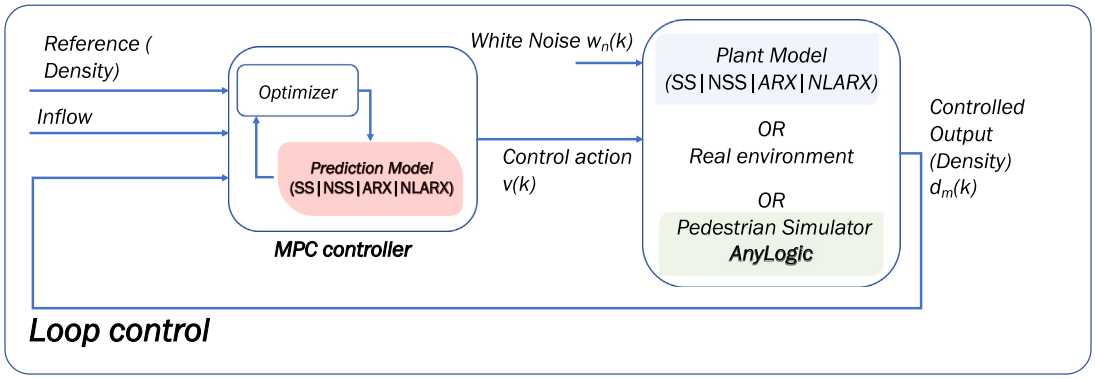

Fig. 1. MPC loop control architecture.

1. Introduction

A critical aspect of emergency evacuations is the efficient control of pedestrian flows, especially in large venues (Helbing and Mukerji, 2012), where the main goals are to ensure the safety of the evacuees by minimizing the risks and enhancing egress operation and comfort.

Technology-based solutions have emerged as promising tools to aid in controlling pedestrian flows (Lopez-Carmona and Paricio-Garcia, 2020). Commonly used solutions include crowd monitoring (Akhter et al., 2019), guidance (Gan et al., 2022) and communication systems (Gorbil and Gelenbe, 2011; van der Wal et al., 2021; Zhang et al., 2017; Zhang and Jia, 2021). Integrating these technology-based solutions to provide real-time instructions represents an opportunity for the actual deployment of interventional mechanisms for regulating pedestrian flows (Haghani, 2020; Haghani and Lovreglio, 2022). It implies developing feedback control systems that monitor the environment and dynamically generate and communicate real-time instructions or recommendations, including exits (Haghani and Sarvi, 2017, 2019b; Lopez-Carmona and Paricio Garcia, 2022, 2021), routes (Wang et al., 2015), schedules (Abdelghany et al., 2014; Murakami et al., 2020), speed, or distance-keeping (Lopez-Carmona, 2022).

Various control technologies may be suitable for implementing feedback controllers in a general context. Among others, Proportional Integral Derivative (PID) control (Ogata, 2010), Fuzzy Logic Control (FLC) (Li and Ranjitkar, 2015), Multi-Agent Systems (MAS) (Liao et al., 2019; Berceanu et al., 2023) or Reinforcement Learning (RL) (Lewis et al., 2012), can be valid alternatives. Another exciting approximation is to use heuristic functions based on discrete-choice modeling theory (Train, 2009), as in Lopez-Carmona and Paricio Garcia (2021), where an adaptive guidance system is proposed that generates optimal exit-choice recommendations based on a discrete-choice model trained through simulation-optimization. However, we would like to highlight Model Predictive Control (MPC) (Camacho and Alba, 2013; Sirmatel and Geroliminis, 2018), which uses a dynamic system prediction model to generate optimal control actions.

MPC control is a particularly suitable approach for regulating pedestrian flows because it may consider the dynamic behavior of pedestrians and the environment (Lopez-Carmona, 2022). It uses a mathematical model to predict the system's future behavior and calculates the optimal control actions that achieve the desired behavior while considering constraints and objectives (see Figs. 1 and 2). Pedestrian flows can be unpredictable and complex, and traditional control methods may not be able to handle the nonlinearities and uncertainties in such systems. MPC can deal with these challenges by continuously updating its model and adjusting its control actions accordingly (Schwenzer et al., 2021). This makes it more adaptable and robust in handling changes in the pedestrian environment, such as sudden changes in pedestrian behavior, new obstacles, or changes in the environment's layout.

Another advantage of MPC over other control system mechanisms is its ability to handle constraints applied to control actions and objectives. In pedestrian flow control, constraints can be applied to limit speed or distance values or their change ratio. MPC can incorporate these constraints into its optimization problem, ensuring that the control actions remain within safe and feasible limits. Finally, MPC can handle multiple objectives simultaneously, including maximizing the flow rate, minimizing the waiting time, and maintaining a safe and comfortable pedestrian environment. It can balance these objectives and optimize the control actions to achieve a desired trade-off. Overall, using MPC for pedestrian flow control offers several advantages, including adaptability, robustness, constraint handling, and multi-objective optimization.

The main component of an MPC controller is the prediction model of the system dynamics, which is used to forecast future system behavior and evaluate the performance of candidate control actions (Fig. 2). Its accuracy and complexity can significantly impact the effectiveness and robustness of the controller (Liu et al., 2022). A poor prediction model can result in incorrect control actions, leading to system instability or poor performance. In contrast, a good prediction model can improve the efficiency and safety of the system, reduce the need for manual intervention, and increase the system's adaptability to changing conditions. Therefore, selecting an appropriate prediction model is crucial in developing an effective and reliable MPC, which will depend on the complexity and heterogeneity of the evacuation environment, the available data, and the performance objectives. One alternative could be to use simulation-based models, but they do not help control pedestrian flows in real-time because they may require a lot of computational

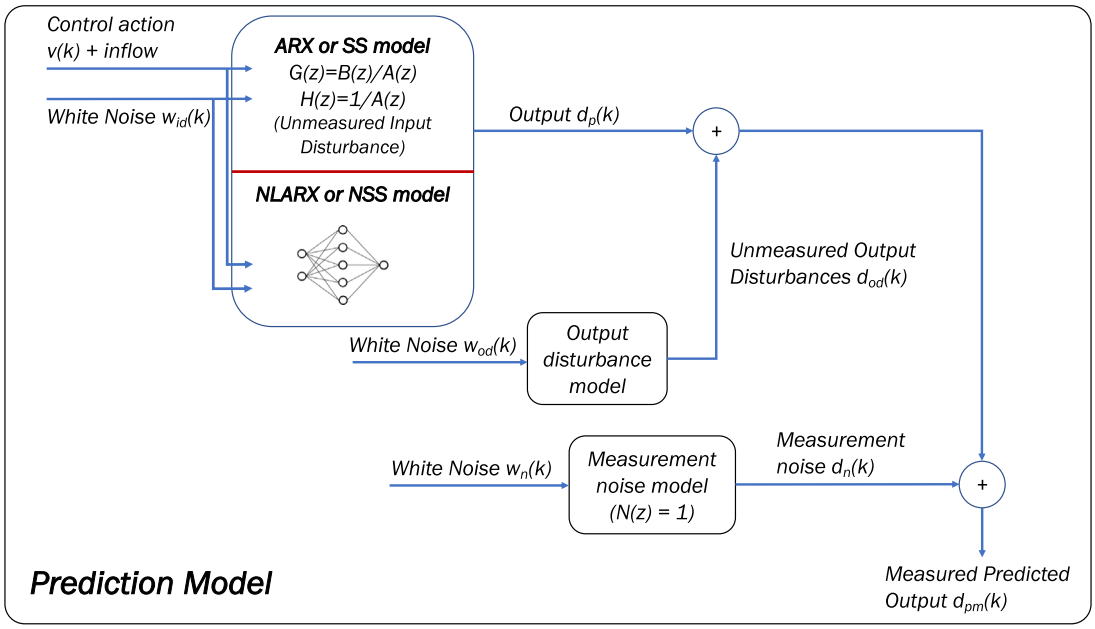

Fig. 2. Prediction model architecture for MPC control.

resources to simulate the system's future behavior iteratively (Ascione et al., 2015). As a result, simplified or data-driven models that require fewer computational resources are preferred for real-time control design in pedestrian flow regulation. These models can provide accurate predictions of the system behavior and can be used when the pedestrian dynamics are unknown or difficult to model analytically (Ghadami and Epureanu, 2022).

In data-driven models, data is collected from the real or simulated system to learn a mathematical model to predict future system behavior. This methodology is named system identification (Ljung et al., 2020), which integrates a model, and the corresponding identification technique to estimate the model's parameters. Standard models used in system identification are transfer functions, state-space models, autoregressive models (input–output models), neural networks, or fuzzy logic models (Ljung, 1999). However, one advantage of using state-space or autoregressive models instead of machine learning-based models is interpretability. State-space and autoregressive models provide clear insight into the system's internal states and dynamics, making it easier to understand how the system works and make informed decisions based on the model's predictions. Additionally, state-space and autoregressive models are generally more computationally efficient than machine learning-based models, which can be important for real-time control applications. In contrast, machine learning-based models may offer better accuracy and predictive power, especially for highly non-linear systems or when dealing large amounts of data.

Existing research exploring the application of the system identification methodology in predicting crowd dynamics for MPC control is limited to identifying Single-Input-Single-Output (SISO) AutoRegressive eXogenous (ARX) models (Lopez-Carmona, 2022). An open research question is if state-space models and their nonlinear counterparts, nonlinear ARX (NARX) (Singh and Sznaier, 2021), and neural state-space models (van Lint et al., 2002) are viable candidates in this application domain. Thus, our study focuses on developing linear and nonlinear prediction models that can be used to design model predictive controllers to regulate the movement of people using instructions that modulate the pedestrians' speed and control congestion. Also, we propose Multiple-Input-Single-Output (MISO) prediction models that use control actions (speed recommendations) and pedestrian inflow measurements as inputs and the local density of pedestrian outflow as output. Input–output datasets are obtained from strategically designed microscopic evacuation simulations based on the Social Force and Agent-Based models over a hybrid simulation framework (Helbing et al., 2000). Several data-driven estimation algorithms are used to identify linear and neural state-space and input–output models, including the subspace method, prediction error minimization, regularized ARX model reduction, and neural network training. We aim to compare and evaluate the models' prediction error performance and quality metrics. Finally, the study investigates the importance of measuring and modeling pedestrian inflow by exploring the case of Single-Input-Single-Output (SISO) systems, where the pedestrian inflow is considered an unmeasured input disturbance. Overall, the paper seeks to contribute to developing efficient pedestrian flow control mechanisms that ensure attendee safety and enhance their experience in large-scale events such as concerts, sporting events, or festivals.

This paper contributes to the active research line on interventional mechanisms that seek to optimize evacuations dynamically rather than describe behavior in evacuations (Haghani, 2020). We aim to fill a gap in the existing literature about implementing crowd dynamics predictors to build MPC controllers:

- It proposes a comprehensive methodology to model crowd dynamics built upon linear and nonlinear system identification, suited to design the MPC controllers.
- Unlike existing studies focused on predicting dynamics using a linear ARX model, we investigate, evaluate and compare linear and nonlinear models, including State-space, Neural State-space, and Nonlinear ARX models.
- It provides insights into how the extension of SISO modeling to MISO modeling influences the predictors' performance.
- It helps better understand how different prediction models perform in terms of prediction error and quality metrics.
- This study suggests that linear state-space models are the most suitable option in a corridor-type scenario and that including pedestrian inflow measurements significantly impact prediction performance.

The paper is organized as follows. Section 2 presents the evacuation simulation framework, providing details on the crowd dynamics simulation model and the density-based metric used to measure the system output. The simulation framework will generate the input–output datasets, including the exogenous control actions, pedestrian inflow measurements, and density measurements at the output area. The system identification methodology and how the datasets were generated are presented in Section 3. Sections 3.2 and 3.3 present the state-space and ARX model identification techniques for predicting crowd dynamics, the corresponding experimental evaluation, and comparisons. Section 4 studies the impact of using SISO models instead of MISO models. The last section provides the discussion, concluding comments, and possible research extensions.

2. Evacuation simulation framework

Microscopic simulations are commonly used in research related to crowd behavior and evacuation (Bi and Gelenbe, 2019). These simulations provide detailed information about how individuals behave during emergencies and evacuations while maintaining safety and keeping costs low (Lovreglio et al., 2020). There are several types of microscopic simulation models used for evacuation analysis, including the Social Force Model (SFM) (Helbing and Molnár, 1995), Cellular Automata (CA) model (Pelechano and Malkawi, 2008; Feliciani and Nishinari, 2016), Agent-Based Model (ABM) (Pan et al., 2007), Fluid-dynamics and lattice gas models (Li et al., 2019), and Flow network models (Liu et al., 2021). Each model has its strengths and is suited for different evacuation scenarios. There is also a growing trend to use data-driven models that learn from real data using neural networks and other machine learning techniques (Yao et al., 2020). Finally, hybrid models that combine different simulation models are becoming increasingly popular for simulating complex evacuation scenarios.

For modeling crowd dynamics in our evacuation scenario, we adopted a hybridization of the SFM (Helbing et al., 2000, 2005) and ABM built as a microscopic simulation framework with the software packages AnyLogic[1] and Matlab.[2] In Lopez-Carmona and Paricio Garcia (2021), the implementation details and motivations to use this framework can be found. AnyLogic integrates the SFM model, which simulates crowd motion, provides density-based measurements, and applies control actions received from Matlab.

Before describing Hebling's SFM model, we should mention that SFM provides a simplified approximation of crowd behavior, rather than presenting an exhaustive model of real-world crowds at a microscopic level. Its effectiveness is highly dependent on the following factors: evacuation scenario, characteristics of pedestrian flows (Lakoba et al., 2005), and the validation and calibration of model parameters to match real-world observations, which is a challenging problem (Johansson et al., 2014). The complexity of crowd behavior and the lack of comprehensive real-world data in controlled evacuation scenarios make it challenging to accurately represent a wide range of crowd scenarios. As a result, the predictive capabilities of this model may have some uncertainties and limitations, mainly at a microscopic level (Chraibi, 2014). However, while it may not capture all the intricacies at the microscopic level of real crowds, it provides insights into flow patterns and emergent behaviors, enabling comparative analysis of different control strategies and prediction models. Instead of calibrating the model with existing evacuation data for a specific scenario, we preferred being as generic as possible, applying a parameterization of the SFM model with typical values and populational variation.

We briefly recall Helbing's SFM used, in which pedestrians i of mass m have an objective speed $v_i^{preferred}$ into a direction $\mathbf{e}_i$, such that their actual speed $\mathbf{v}_i$ adapts with a characteristic time τ_i. The following force equation gives the change of velocity:

$$\mathbf{f}_i(t) = m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{v_i^{preferred}(t)\mathbf{e}_i^{preferred}(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{i \neq j} \mathbf{f}_{ij} + \sum_{O} \mathbf{f}_{iO}\,, \tag{1}$$

where $\mathbf{f}_{ij}$ and $\mathbf{f}_{iO}$ represent repulsive forces to other pedestrians and obstacles. Thus, from Eq. (1) pedestrian speed is derived as

$$\mathbf{v}_i(t+\triangle t) = \mathbf{v}_i(t) + \frac{\mathbf{f}_i(t)}{m_i}\triangle t\,.$$

The force $\mathbf{f}_{ij}$ is modeled by

$$\mathbf{f}_{ij} = A_i exp\left[(r_{ij} - d_{ij})/B_i\right]\mathbf{n}_{ij} \cdot \Omega(\lambda_i, \varphi_{ij}) + kg(r_{ij} - d_{ij})\mathbf{n}_{ij} + \kappa g(r_{ij} - d_{ij}) \triangle v_{ji}^t \mathbf{t}_{ij}\,, \tag{2}$$

where $A_i[N]$ and $B_i[m]$ represent the strength and range of interaction between pedestrians, $r_{ij} = (r_i + r_j)$ is the sum of the pedestrians' radii, $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ is the distance between them, $\mathbf{n}_{ij}$ is the unit vector pointing from j to i, and function $\Omega(\lambda_i, \varphi_{ij}) \Rightarrow [0, 1]$ represent the anisotropy of interactions, used to weigh the psychological repulsive forces considering visual angle:

$$\Omega(\lambda_i, \varphi_{ij}) = \lambda_i + (1 - \lambda_i)\frac{1 + cos(\varphi_{ij})}{2}$$

[1] https://www.anylogic.com/ Accessed 10 Feb 2023.

[2] https://www.mathworks.com/ Accessed 10 Feb 2023.

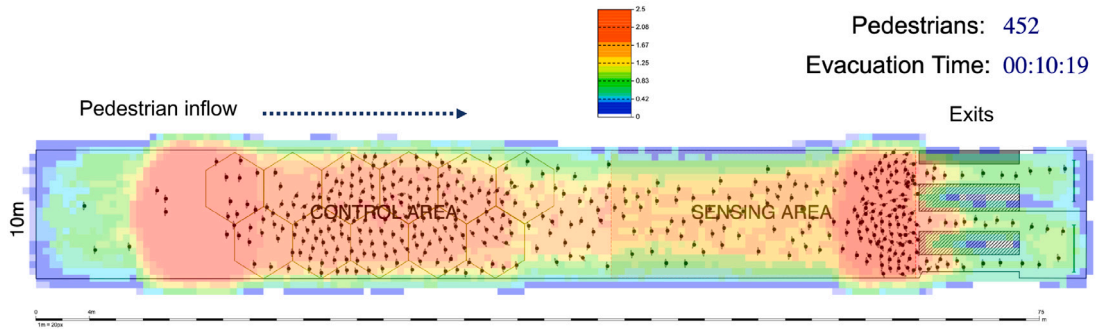

Fig. 3. Schematic for the evacuation scenario. Incoming pedestrian flow from the left is evacuated through the exits. The cells represent the control area, while the sensing area at the exit gates provides real-time density measurements.

where φ_{ij} denotes the vision angle, and $0 \leq \lambda_i \leq 1$ defines the influence area.

When pedestrians touch each other, the second and third terms in Eq. (2) are considered:

- A body force $kg(r_{ij} - d_{ij})\mathbf{n}_{ij}$ counteracting body compression.
- A sliding friction force $\kappa g(r_{ij} - d_{ij}) \triangle v^t_{ji}\mathbf{t}_{ij}$ counteracting relative tangential motion $\mathbf{t}_{ij} = (-n^y_{ij}, n^x_{ij})$, where $\triangle v^t_{ji} = (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{t}_{ij}$ denotes the difference in tangential with velocity.

If pedestrians do not touch each other, $g(x) = 0$, otherwise $g(x) = x$. The interaction with obstacles $\mathbf{f}_{iO}$ is analogous by substituting pedestrian j with the closest point to the obstacle.

Another crucial aspect of the model is how the preferred direction is managed, which may significantly impact pedestrian flow through bottlenecks. Given the relative simplicity of the evacuation layout and to be as generic as possible, we applied a shortest-path approach instead of other options such as herding or least-congested path (Haghani and Sarvi, 2019a). When injecting pedestrians into the system, we randomly assigned coordinates to each agent's destination (within the vertical line behind the three exits). Then, at each simulation step, agents update the preferred direction $\mathbf{e}_i^{preferred}$ to point to their destination coordinates.

It must be pointed out that while considering the shortest path is a common approach to modeling pedestrian route choice, several alternative mechanisms can be used to capture the desired direction when simulating pedestrian motion in more complex environments with multiple exits and alternative paths. These mechanisms consider many factors influencing pedestrians' decision-making processes (Bode et al., 2015). We give three examples of alternative or complementary mechanisms to elicit the desired direction without being exhaustive. Utility-based models involve assigning utility values to different routes based on factors such as travel time, distance, the attractiveness of the surroundings, social factors, and individual preferences (Lopez-Carmona and Paricio Garcia, 2022; Haghani et al., 2014; Lovreglio et al., 2016). Pedestrians then choose the route that maximizes their perceived utility. Cognitive models focus on the decision-making processes of individuals and attempt to simulate pedestrian behavior based on cognitive factors, such as memory, perception, learning, and adaptation (Zia and Ferscha, 2020; van der Wal et al., 2017). These models consider how pedestrians perceive and process information to decide their desired direction. Finally, risk-averse models account for pedestrians' risk perception and hazard aversion (Mohd Ibrahim et al., 2017). Pedestrians may choose routes that minimize exposure to potential risks or hazardous situations, such as busy roadways or congested areas.

In this work, we have used the same parameterization as in Lopez-Carmona (2022). Pedestrians have uniformly distributed masses in the interval [65 Kg − 85 Kg], the default preferred speed is Gaussian distributed with a mean value of 1.3 m/s and a standard deviation of 0.3 m/s representing an urgent situation, and the acceleration time τ_i is distributed with a mean of 1 s and a standard deviation of 0.2. As proposed in Helbing et al. (2000), $k = 1.2 \cdot 10^5$ kg s^{-2}, $\kappa = 2.4 \cdot 10^5$ kg m^{-1}s^{-1}, $A_i = 2 \cdot 10^5$ N, $B_i = 0.08$ m, while the influence area is set to $\lambda_i = 0.1$.

Our evacuation scenario is a 75 m long and 10 m wide corridor with three exit gates (Fig. 3). A variable incoming pedestrian flow $q(t)$ traverses a control area where control actions in the continuous domain (speed instructions) $s(t)$ are communicated to pedestrians, who adapt their preferred speed $v_i^{preferred}$ according to the formula $s(t) * n(1, 0.3, 0.6, 1.4)$. $n()$ represents a truncated normal distribution with mean 1, standard deviation 0.3, and valid range of values 0.6 − 1.4. This formula accounts for the stochastic response of pedestrians to speed instructions.

Finally, the sensing area downwards provides periodic density measurements $d(t)$, widely used to estimate inherent risk and performance in pedestrian crowds (Feliciani and Nishinari, 2018). The local density is estimated using Voronoi diagrams (Steffen and Seyfried, 2010; Zhang and Seyfried, 2013). Being Ω_i the area available to the pedestrian, the pedestrian-specific density is defined as:

$$\rho_i(t_k) = \frac{1}{|\Omega_i(t_k)|},$$

and the average density at time t_k is:

$$d(t_k) = \frac{1}{n}\sum_{i=1}^{n} \rho_i(t_k),$$

where n is the population of agents.

3. System identification for predicting crowd dynamics

System Identification "deals with the problem of building mathematical models of dynamical systems based on observed data from the system" (Ljung, 1999; Ljung et al., 2020). The built models can then be used for a variety of purposes, such as control, forecasting, and analysis of the system's behavior.

Numerous methods can be employed to identify systems by learning the dynamic correlations among measured variables. These approaches enable the development of various process models, including transfer functions, linear and nonlinear autoregressive models, and state-space models. To estimate the dynamics of nonlinear systems, one can use Hammerstein–Wiener and nonlinear ARX models alongside machine learning techniques such as Gaussian processes, support vector machines, neural networks, tree ensemble learning, and other nonlinear representations. Additionally, it is possible to generate neural models of ordinary differential equations by employing deep learning methods, which can capture the dynamic behavior of nonlinear systems.

A system identification workflow typically includes the following tasks[3]: generation of input–output datasets, linear and nonlinear model identification, model validation, and use of the identified model in control system design, online estimation, simulation, and prediction. In the following, we describe the tasks accomplished to generate the input–output datasets used to estimate and validate the prediction models. Then, the linear and neural state-space models are identified and validated, and finally, the corresponding ARX counterparts.

Our work focuses on the practical design of interventional mechanisms using feedback controllers rather than offline route optimization or prediction models for non-controlled environments. We aim to develop practical surrogate models to be used as predictors in Model Predictive Control (MPC) controllers to design adaptive guidance systems (see Fig. 1).

Real crowd data is intuitively the first choice for data-driven estimation of prediction models (see outstanding experiment repository with real crowds in Re3data.Org (2020) and Boomers et al. (2023)). However, we have opted for synthetic pedestrian flow data and a generic configuration of the pedestrian simulator for the following reasons. Building an MPC prediction model requires considering control inputs and corresponding outputs, which implies generating new input–output datasets in a real environment, requiring numerous evacuation experiments and variations that would need to be applied to the inputs and outputs of the system, which has safety and cost implications. Considering that our main goal is to test the relative performance of several prediction models, we found it reasonable to use synthetic data instead. Using this approach, the final task would be to test and validate with real crowds the MPC controller equipped with the obtained prediction models. In this paper, we are limited to obtaining prediction models to be used within MPC controllers.

3.1. Generation of datasets

For generating the input–output datasets, we performed 200 random evacuation simulations. In each evacuation simulation of one-hour duration and a sampling period of $T = 2$ s, a variable pedestrian flow $q(k)$ was injected in the evacuation scenario, defined as a random Gaussian signal in the range $0 - 125$ peds/min ($[\mu - \sigma,\ \mu + \sigma]$). A 1/60 Hz cutoff low-pass filter was applied to this signal to set a reasonable limit on the variation of the pedestrian inflow. On the other hand, in the control area, a speed control signal $s(k)$ was applied as a random Gaussian signal in the range $0.05 - 1.36$ m/s ($[\mu - \sigma,\ \mu + \sigma]$). At each 2 s period, the density measurements were recorded in signal $d(k)$. Thus, $q(k)$ and $s(k)$ input signals and output $d(k)$ define a multiple-input single-output dynamic system. Fig. 4 represents a single-shot evacuation simulation experiment, where Inflow and Speed subplots are the inputs, and the Density subplot at the top shows the system output.

From the 200 experiments, the first 100 experiments were used for estimating the models, and the last 100 experiments for validation tasks. It is worth noting that when estimating models from multi-experiment datasets, we may merge them by averaging out independently estimated models or group them in a single dataset. Since the conditions in the different experiments were about the same, we merged the data since it typically involves better-conditioned calculations and more efficiency.

3.2. Linear state-space and neural state-space models

The variables describing our system are discrete-time signals, so we focused on discrete-time state-space models, which describe a dynamic system using first-order difference equations. In linear state-space models, the difference equations are linear in the state and input variables. Thus, the discrete-time linear state-space model structure in the *innovations* form including noise is structured as:

$$x(k+1) = Ax(k) + Bu(k) + Ke(k)$$
$$y(k) = Cx(k) + Du(k) + e(k)$$

where k represents a discrete-time instant, $x(k)$ represents the vector of states (its length is the model order), $u(k)$ is the input vector $\{q(k), s(k)\}$, and $y(k)$ is the output density $d(k)$. In the innovations form, the state-space equations include a single noise source $e(k)$, instead of separate process and measurement noise. The matrixes A, B, C, D and K are the numerical structures to be estimated.

[3] https://www.mathworks.com/help/ident/gs/system-identification-workflow.html.

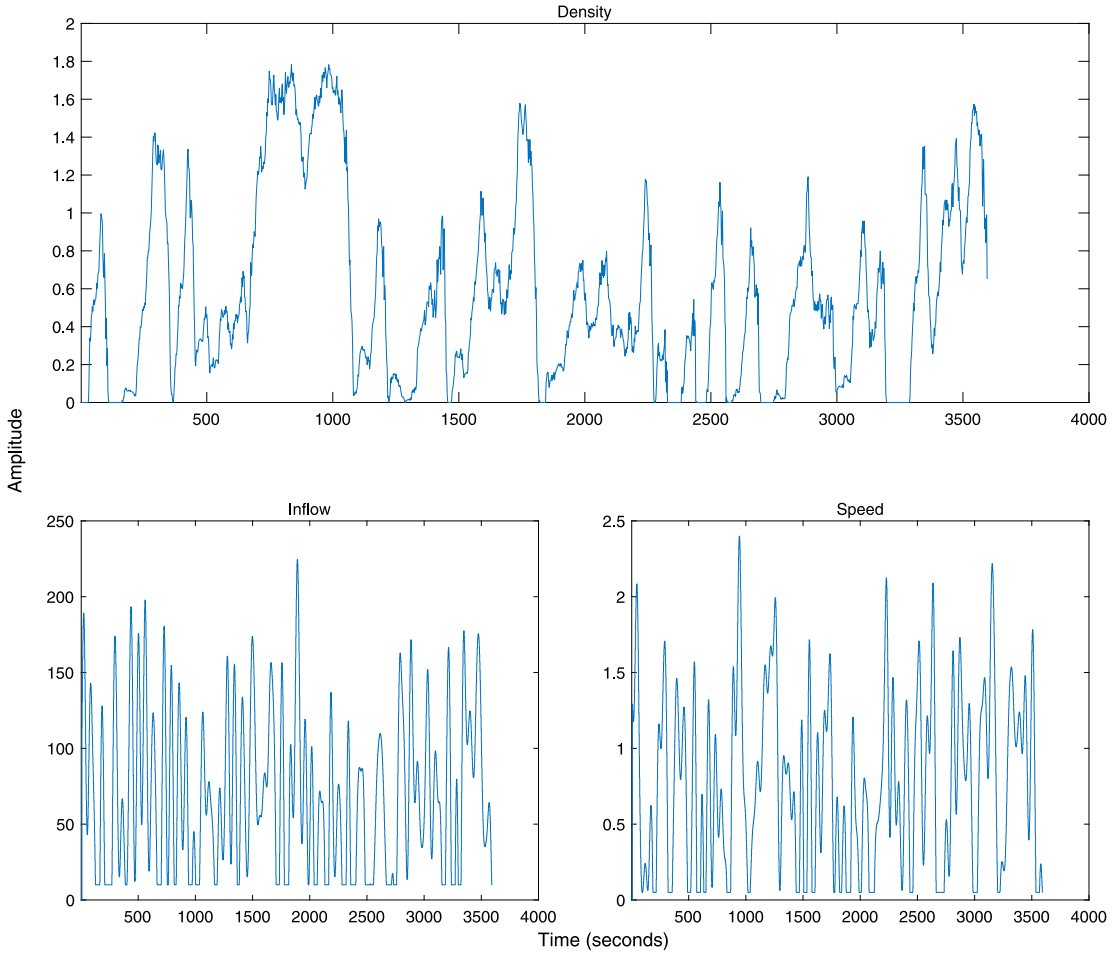

Fig. 4. Example of input–output ($[Inflow, Speed] - Density$) data for a single experiment. Inflow $q(k)$ is measured in peds/min, speed $s(k)$ in m/s and density $d(k)$ in peds/m^2.

Note that state-space representation in discrete-time is equivalent to the linear Autoregressive Moving Average with Extra Input model (ARMAX), rearranged to include only one delay in the state equations.

We assumed no prior knowledge of the crowd dynamics, so we took a black-box estimation approach with free parameterization, where the error to be minimized was the one-step-ahead prediction error between the measured and predicted outputs. It means that the estimation aims to produce a good predictor model. All the elements in A, B, C, D, and K were automatically adjusted by the estimation algorithms guaranteeing well-conditioned calculations. For each estimation algorithm, we picked the best model order from the range $n = 1 - 10$ measuring each state's contribution, calculated as the logarithm of singular covariance matrix values. Only the time delays of the inputs (transport delays) were set before all the identification processes. These delays were obtained by comparing the fitness of a predefined AutoRegressive eXogeneous (ARX) model to the estimation datasets, varying the delays of $q()$ and $s()$. Thus, given the ARX structure:

$$y(k) + a_1 y(k-1) + \cdots + a_{na} y(k-na) =$$
$$b_1 u(k-nk) + \cdots + b_{nb} u(k-nb-nk+1) + e(k)$$

where the model order was set to $na = nb_{q()} = nb_{s()} = 2$, and nk is the input delay, we obtained the best fitness for time delays of 60 s ($nk_{q()} = 30$) and 28 s ($nk_{s()} = 14$) for $q()$ and $s()$ respectively. These values are consistent with the topology of the evacuation scenario, the average pedestrians' speed, and the location in which the inputs are applied.

We first estimated linear state-space models with the noniterative *subspace method* (SID) (Van Overschee and De Moor, 1996), which uses a weighting scheme for singular-value decomposition (SVD). We tried three different algorithms, the Canonical Variate Algorithm (CVA) (Larimore, 1990), MOESP (Verhaegen, 1994), and ARX-based estimation (Jansson, 2003). We observed slightly better quality metrics for the CVA and MOESP algorithms. Thus, we selected the model obtained using the most widely used CVA algorithm (see Table 1).

Table 1
Linear state-space models obtained with the subspace identification and prediction error minimization methods. The input delays in samples are $nk = [30, 14]$.

Subspace Identification Method (SID)				Prediction Error Minimization (PEM)			
A	=			A	=		
	x1	x2	x3		x1	x2	x3
x1	0.9877	−0.09357	0.04712	x1	1.014	0.09384	−0.04235
x2	0.0319	0.8275	0.2683	x2	−0.04202	0.8489	0.295
x3	−0.0115	0.03344	0.8962	x3	0.003087	0.01274	0.9013
B	=			B	=		
	Inflow	Speed			Inflow	Speed	
x1	−1.24e−06	−0.0002314		x1	−0.0001138	−0.0003881	
x2	−9,69e−07	−0.0004993		x2	1.245e−05	0.0007906	
x3	1.18e−06	4.97e−05		x3	−5.044e−05	−4.306e−05	
C	=			C	=		
	x1	x2	x3		x1	x2	x3
Density	−138.9	2.735	0.2013	Density	−64.32	−1.295	−0.4819
D	=			D	=		
	Inflow	Speed			Inflow	Speed	
Density	0	0		Density	0	0	
K	=			K	=		
	Density				Density		
x1	−0.006544			x1	−0.01505		
x2	0.01239			x2	−0.01949		
x3	−0.004544			x3	0.009676		

Next, we estimated a model using the *prediction error minimization* (PEM) approach (Ljung, 1999). This method initializes the parameter estimates using the *subspace method* and refines their values iteratively by minimizing the prediction error using the least square criterion. The optimization method used for the iterative parameter estimation was a combination of search methods tried in sequence: *subspace Gauss–Newton*, *adaptive subspace Gauss–Newton*, *Levenberg–Marquardt*, and *steepest descent*. The first solution leading to a reduction in prediction error was used at each iteration of the algorithm. The model is shown in Table 1.

Finally, we performed state-space estimation by *reduction of a regularized ARX model* (REG), which is typically less sensitive to options than the subspace method. The regularized ARX model structure was set with order $na = 10, nb_{q()} = nb_{s()} = 70$, using QR factorization for solving the overdetermined set of linear equations defining the least-squares problem, and a tuned and correlated kernel for regularization (Chen et al., 2012). The state-space matrixes were calculated from the estimated ARX model, discarding the states with the smallest Hankel singular values. When discarding the specified states, the remaining states were not altered. The obtained state-space model is presented in Table 2.

For illustration purposes, Fig. 5 shows the simulation and prediction responses of the linear models estimated for the input and output signals of one of the experiments recorded in the validation dataset. Each graph shows the actual measured output signal and the output generated by each model. The numerical values in the legends represent NRMSE fitness value in percentage

$$fit = 100(1 - \frac{\|d - \hat{d}\|}{\|d - mean(d)\|}),$$

where d is the measured density, and $\hat{d}$ is the output of the model.

Fig. 6 shows the aggregated NRMSE fitness results for the 100 experiments in the validation set. Each graph shows the median boxplots for the estimated models, SID, PEM, and REG, and each type of response generated: simulation and predictions at $5(10s)$, $15(30s)$, $30(60s)$, $60(120s)$, and $90(180s)$ steps ahead. Interestingly, we observed better fitness with the PEM and REG algorithms. At the same time, the quality metrics shown in the Table 3: *FPE (Akaike's Final Prediction Error)*, *AIC (Akaike's Information Criterion)*, *nAIC (Normalized AIC),* and *BIC (Bayesian Information Criterion)* computed as properties of the output disturbance $e(k)$, confirm the PEM-based model as the best choice to compare to nonlinear approaches.

A natural nonlinear counterpart of the state-space model is the neural state-space model (NSS), in which the functions that account for nonlinear dynamics are neural networks. An NSS model in the discrete domain can be defined as follows,

$$\begin{aligned} x(k+1) &= \varphi(x(k), u(k), v_1(k); \theta_\varphi) \\ y(k) &= \omega(x(k), v_2(k); \theta_\omega), \end{aligned} \tag{3}$$

where functions φ and ω are the state and nontrivial output functions approximated by neural networks, and $v_1(k)$ and $v_2(k)$ are sequences of zero mean identically independent random vectors. The specificity of this model, in contrast to input–output models such as ARX or nonlinear ARX, arises from the fact that only the outputs have desired values, while the internal state variables do not need to be imposed. This characteristic gives state-space models more flexibility, providing the ability to model more complex

Table 2
Linear state-space model obtained with the reduction of a regularized ARX model method.

Reduction of regularized ARX model identification method (REG)										
A	=									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	0.9812	0.3164	−0.03652	0.1307	0.1014	0.03904	−0.04154	−0.08731	0.09128	−0.04591
x2	0.01694	0.9643	−0.03374	0.4888	−0.7875	−0.2391	−0.07976	0.2452	0.2825	−0.1924
x3	−0.0007516	−0.02129	0.9104	0.2	0.08735	−0.1866	0.01862	−0.06976	0.1352	−0.06707
x4	−0.009502	−0.04187	0.1652	0.113	−0.2137	0.5488	0.05345	0.1354	−0.412	0.3002
x5	0.02212	0.2097	−0.0446	0.06205	0.101	0.3964	−0.1387	0.4969	0.2461	0.1056
x6	0.006092	−0.03666	0.2619	−0.5528	−0.4115	0.2864	0.2483	0.08633	−0.02515	−0.1211
x7	−0.01542	0.00438	−0.07138	0.3315	0.254	0.4513	0.7097	−0.3511	0.1707	−0.1194
x8	−0.009216	−0.04623	0.01252	0.004839	−0.04866	−0.1678	0.5445	0.491	−0.5036	−0.06555
x9	0.01761	0.06584	0.1067	0.00779	−0.3146	0.05527	0.07155	−0.008003	0.06007	0.6851
x10	−0.01176	−0.044	−0.09925	−0.109	0.1556	0.04498	0.1212	−0.2254	−0.4552	0.6401
B′	=									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Inflow	−0.0001441	0.00206	0.0004501	0.000342	0.001801	0.001161	−0.001014	0.001426	−0.003234	0.004704
Speed	0.05892	−0.2185	−0.06392	0.07388	−0.1766	−0.05611	0.07529	0.09376	0.149	−0.0004222
C	=									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Density	1.724	0.4556	−0.113	0.5461	−0.5629	0.02295	−0.1043	−0.1536	0.004233	−0.01477
D	=									
	Inflow	Speed								
Density	0	0								
K′	=									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Density	0.4719	0.3476	−0.006503	0.02065	0.01752	−0.02262	−0.005738	−0.04775	0.004882	0.009913

Table 3
FPE (Akaike's Final Prediction Error), *AIC (Akaike's Information Criterion)*, *nAIC (Normalized AIC)*, and *BIC (Bayesian Information Criterion)* quality metrics of the linear identified models.

Linear state-space models' quality metrics			
	SID	PEM	REG
FPE	0.0033	0.0032	0.0032
AIC	−5.1868e+04	−5.1997e+04	−5.1837e+04
nAIC	−5.7227	−5.7298	−5.7209
BIC	−5.1541e+04	−5.1904e+04	−5.0746e+04

behaviors. Also, it is still possible to define states as observable variables by including them as direct outputs in the output equation and the training datasets.

Based on Eq. (3), the one-step-ahead predictions of the state and output can be computed recursively,

$$x_p(k+1) = \varphi(x_p(k), u(k), 0; \theta_\varphi) + K(k, x_p(k))(y_m(k) - \omega(x(k), 0))$$
$$y_p(k+1) = \omega(x_p(k+1), 0; \theta_\omega)\,,$$

where $K(k, x_p(k))$ is the extended Kalman gain, $y_m(k)$ is the measured output, and $y_m(k) - \omega(x_p(k), 0)$ is the prediction error. As the functions φ and ω are not known, the following time-invariant predictor with the more general form can be used,

$$x_p(k+1) = \varphi(x_p(k), u(k), y_m(k); \theta_\varphi)$$
$$y_p(k+1) = \omega(x_p(k+1); \theta_\omega)\,.$$

Here, we may take different approaches to implement the neural predictor. Considering our system has a single observable variable (density), we may remove the neural network w and directly map the predicted state to the predicted output,

$$x_p(k+1) = \varphi(x_p(k), q(k), s(k), d_m(k); \theta_\varphi)$$
$$d_p(k+1) = x_p(k+1)\,.$$

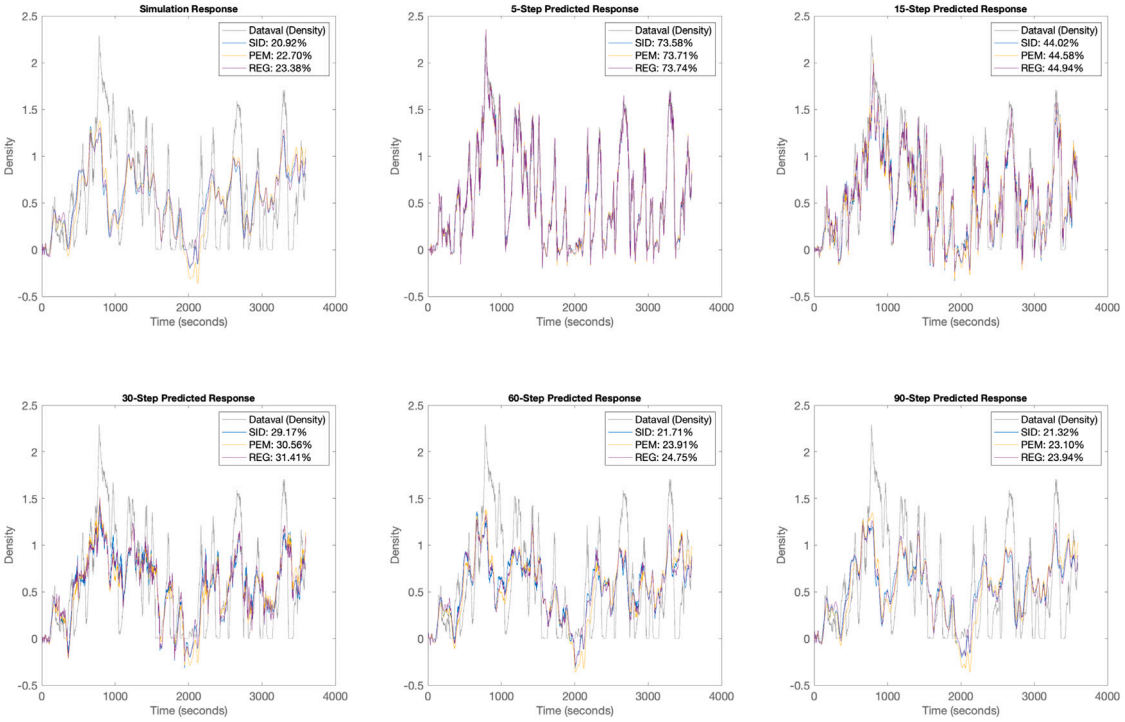

Fig. 5. Simulated and predicted response comparisons (Normalized Root Mean Square Error, NRMSE fitness) to validation data. Dataval (Density) is the measured density. SID, PEM, and REG show the responses for the different linear state-space models: *subspace*, *prediction error minimization*, and *reduction of ARX model* methods.

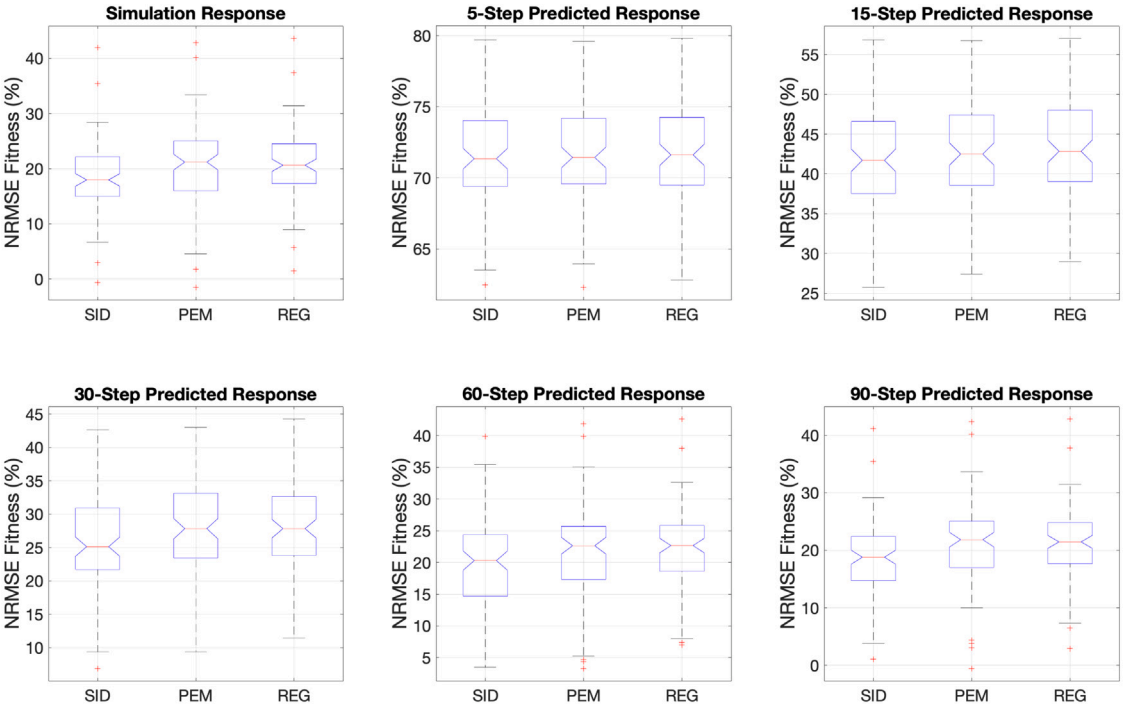

Fig. 6. NRMSE fitness boxplots of simulation and prediction responses of the SID, PEM, and REG linear models.

Table 4
Quality metrics of the neural state-space identified models. The columns [X Y] define the number of layers and neurons per layer. The PEM column represents the quality metrics for the linear state-space model obtained using the prediction error minimization algorithm.

Neural state-space models' quality metrics						
	PEM	[512]	[64 64]	[128 128]	[256 256]	[512 512]
FPE	0.0032	0.0090	0.0064	0.0041	0.0042	0.0038
AIC	−5.1997e+04	−3.3736e+04	−3.9837e+04	−4.7616e+04	−4.7581e+04	−4.9406e+04
nAIC	−5.7298	−4.7142	−5.0535	−5.4861	−5.4842	−5.5857
BIC	−5.1904e+04	−3.3736e+04	−3.9837e+04	−4.7616e+04	−4.7581e+04	−4.9406e+04

Therefore, the identification process consists of estimating the parameters of the neural network θ_φ that minimize the prediction error defined as a cost function, which is to be minimized iteratively,

$$J(\theta_\varphi) = 1/N \sum_{k=0}^{N-1} (d_m(k+1) - d_p(k+1))^2 ,$$

where N is the length of each training sequence.

As is common in deep learning and variational inference tasks, the specific choices of neural network architecture and hyper-parameters can significantly impact the resulting performance and generalization of the model. As long as an architecture solves the problem with minimal computational costs, then that is one that we should use. Thus, we manually tested different configurations of the feedforward neural network φ by varying the number of hidden layers and neurons. Our general strategy was first to try one and two hidden layers and progressively increase the number of hidden neurons. Our priority was to expand the network by adding more hidden neurons. It is worth noting that doubling the size of a hidden layer is less expensive, in computational terms, than doubling the number of hidden layers. In the parameterization of the ADAM training algorithm, we set the maximum number of epochs to 200, the activation function to tanh, and the mini-batch size to 10, being the size of the mini-batch the subset of the training set used to evaluate the gradient of the loss function and update the weights.

As shown in Table 4, the quality metrics for the neural state-space models improved for an increasing number of neurons and layers, though they did not outperform any of the linear state-space models. These results indicate that nonlinear modeling based on the system's internal state for our controlled evacuation scenario does not contribute to an improvement in prediction performance. We could extend the state-space to include several past observable outputs, but we would no longer deal with a reduced-order model, which ultimately represents one of the advantages of state-space modeling. Therefore, we considered evaluating and analyzing input–output modeling approaches, namely linear and nonlinear autoregressive models (ARX and NARX), not based on state-space modeling but on the use of past inputs and past observable outputs.

3.3. Linear and nonlinear AutoRegressive eXogenous models

AutoRegressive eXogenous (ARX) models (the exogenous variable is the control input) are derived from the more general *polynomial linear model*,

$$A(z)y(k) = \frac{B(z)}{F(z)}u(k-nk) + \frac{C(z)}{D(z)}e(k),$$

where $A(z)$, $B(z)$,... are polynomials in z^{-1} (time-shift z operator), nk is the transport delay, and $e(k)$ is white noise with variance σ_e^2, verifying that the input signal and noise are uncorrelated. In ARX, $F(z) = 1$ and $C(z)/D(z) = 1$, which means that the noise and the dynamics model are coupled. Thus, the corresponding ARX structure can be defined as:

$$A(z)y(k) = B(z)u(k-nk) + e(k) ,$$

or equivalently

$$y(k) = a_1 y(k-1) + \cdots + a_{na} y(k-na) + b_1 u(k-nk) + \cdots + b_{nb} u(k-nb-nk+1) + e(k) .$$

Typically, the parameters are estimated using a prediction error minimization approach with the input–output measured data $u(k)$ and $y_m(k)$, where the prediction error $\varepsilon(k)$ is:

$$\varepsilon(k,\theta) = A^{-1}(z,\theta)[y_m(k) - B(z,\theta)u(k)] ,$$

and the objective is to minimize the cost function $V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(k,\theta)^2$ (least-squares criterion).

Before estimating an ARX model, the most appropriate structure needs to be predefined. For this purpose, all combinations of model orders na and nb in the range $[1-20]$ were first generated. We estimated an ARX model for each combination of model orders and calculated the normalized prediction error to the validation set. Finally, the following best-fitting model was selected according to Akaike's Information Criterion (AIC), with polynomial orders $na = 11$ and $nb = [6, 6]$:

$$A_{density}(z) = 1 - 0.9967z^{-1} - 0.134z^{-2} + 0.01211z^{-3}$$

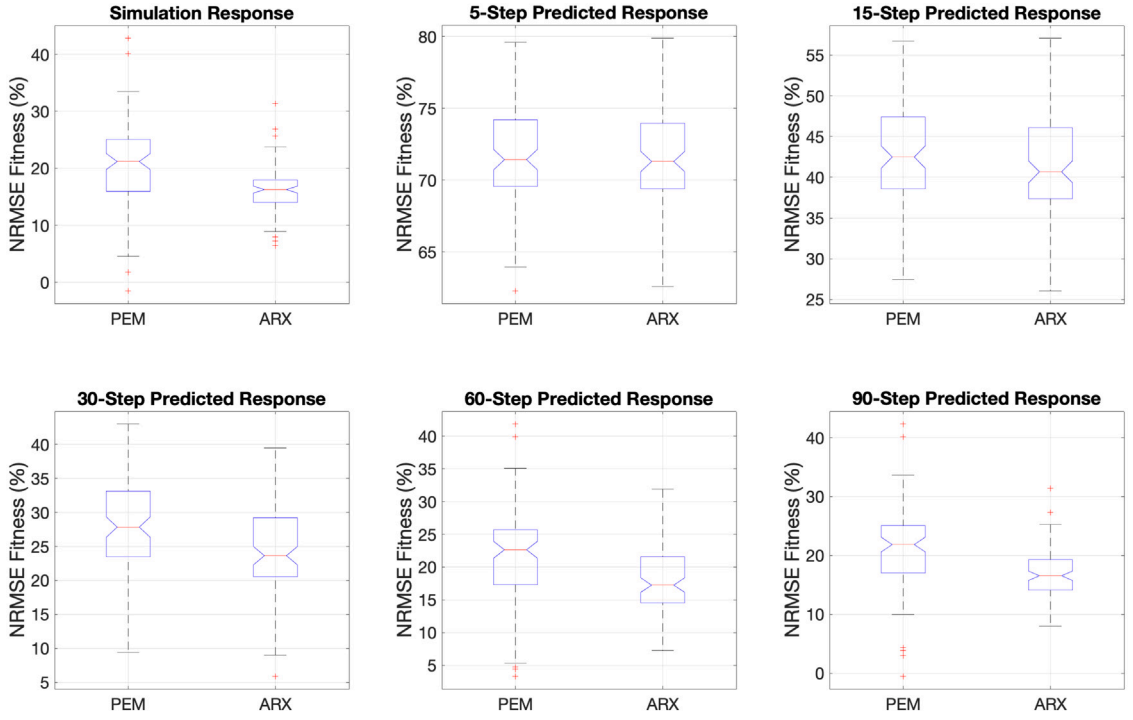

Fig. 7. NRMSE fitness boxplots of simulation and prediction responses of the PEM and ARX linear models.

Table 5
Quality metrics of the linear state-space and ARX models.

Linear state-space and ARX models' quality metrics		
	PEM	ARX
FPE	0.0032	0.0033
AIC	−5.1997e+04	−5.1759e+04
nAIC	−5.7298	−5.7166
BIC	−5.1904e+04	−5.0722e+04

$$
\begin{aligned}
&+ 0.01374z^{-4} + 0.06025z^{-5} + 0.05216z^{-6} \\
&+ 0.019z^{-7} - 0.01408z^{-8} + 0.02011z^{-9} \\
&- 0.01971z^{-10} + 0.0007183z^{-11} \\
B_{inflow}(z) &= 0.02725 - 0.02314z^{-1} - 0.006962z^{-2} \\
&- 0.0008957z^{-3} + 0.003591z^{-4} + 0.006749z^{-5} \\
B_{speed}(z) &= 0.06234 - 0.05255z^{-1} - 0.008939z^{-2} \\
&- 0.0147z^{-3} + 0.04313z^{-4} - 0.02863z^{-5} .
\end{aligned}
$$

Table 5 compares the linear state-space and ARX models' quality metrics, showing that the linear state-space modeling outperforms ARX. Interestingly, the median boxplots of NRMSE fitness for PEM and ARX (Fig. 7) demonstrate that the linear state-space modeling is significantly better at increasing step-ahead predicted responses and at simulation, even though the order of the state-space model is 3 in contrast to the order 11 of the ARX model. It indicates that modeling the inner state of the evacuation process has a positive influence on modeling the system dynamics.

As with neural state-space models, we can also identify nonlinear ARX models (NARX), corresponding to the following predictor,

$$y_p(k+1) = f(y_m(k), \ldots, y_m(k-na), u(k-nk), \ldots, u(k-nk-nb+1)) ,$$

such that f is a nonlinear mapping function of passed inputs u and measured outputs y_m (regressors). Thus, the output function is built upon linear, nonlinear, and offset components acting on the regressors. Firstly, the regressor values are obtained, being, in the simplest case, linear delayed inputs and outputs. Then, the regressors are assigned as inputs to the output function's linear and nonlinear blocks in the second stage. Formally, the output function can be defined as:

$$f(r) = L^T(r) + g(Q(r)) + O ,$$

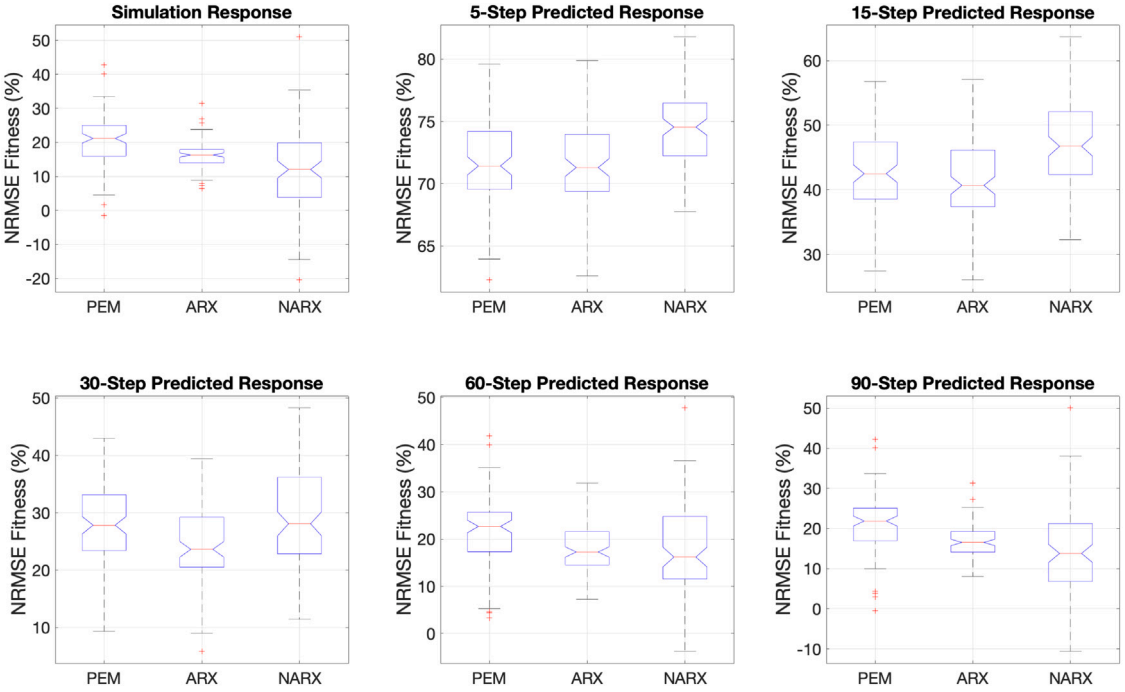

Fig. 8. NRMSE fitness boxplots of simulation and prediction responses of the PEM, ARX and NARX models.

where r is the regressor vector, $L^T(r)$ is the linear function output, $g(Q(r))$ represents the nonlinear function output, where Q is a projection matrix that makes the calculations well-conditioned, and O is an offset. Typical alternatives for the nonlinear functions are *tree-partition, sigmoid,* and *wavelet networks*. On the other hand, we may use machine learning-based functions such as *multilayered neural networks*, *gaussian process regression*, *tree ensemble regression*, or *support vector machine*.

To resemble the neural state-space approach, we focused on *feedforward neural networks* where the inputs were the regressors' values,

$$y_p(k+1) = \varphi(d_m(k), \dots, d_m(k-na), q(k-nk_q), \dots, q(k-nk_q-nb_q+1), s(k-nk_s), \dots, s(k-nk_s-nb_s+1); \theta_\varphi)\,.$$

The corresponding NARX models were estimated using the same regressors as in the best-fitting ARX model previously obtained and optimizing the 1-step ahead predicted response using the Levenberg–Marquardt backpropagation training algorithm. We tried neural networks with one and two layers, and the following layer sizes: $[2, 5, 10, 30, 50, 100]$. We found the best results for two layers and five neurons per layer, representing a significantly smaller layer size than in the neural state-space models (Fig. 8). Also, we observed a similar pattern in all the experiments, with a better prediction fitness up to 15 steps ahead, worsening for increasing values. Interestingly, for short-term predictions, NARX has the best performance compared to PEM and ARX, but PEM is preferred for long step-ahead predictions.

4. Single-Input-Single-Output (SISO) modeling

So far, we studied the identification of predictive models for controlled evacuation scenarios in which real-time measurements of incoming pedestrian flow were available. From a practical perspective, it was found interesting to investigate controlled evacuation scenarios in which pedestrian inflow measurements were unavailable so that these would be part of the unmeasured input disturbances in the prediction models. Thus, we estimated SISO models from the same experimental datasets, using only the input corresponding to the control actions (speed instructions).

We first estimated linear state-space models using the SID, PEM, and REG estimation algorithms and then neural state-space models. As Tables 6 and 7 show, the best performance was obtained with a linear state-space model of order 3 using the prediction error minimization algorithm:

The best neural state-space model found had two layers and 64 neurons per layer, and as with the MISO models, its quality was significantly below the linear state-space models. So, the NSS model was discarded as a potential prediction model.

Finally, we identified linear and nonlinear input–output models, ARX and NARX, following the same methodology as in the MISO models. Thus, we selected the following ARX model with polynomial orders $na = 11$ and $nb = 18$:

$$A_{density}(z) = 1 - 0.9989z^{-1} - 0.1347z^{-2} + 0.01167z^{-3}$$

Table 6
Linear state-space SISO model obtained with the prediction error minimization method. The input delay in samples is $nk = 14$.

Prediction Error Minimization (PEM)			
A	=		
	x1	x2	x3
x1	0.9537	0.09953	0.01412
x2	0.04817	0.5909	−0.3844
x3	−0.03746	−0.1525	0.447
B	=		
	Speed		
x1	0.001305		
x2	−0.006876		
x3	−0.005608		
C	=		
	x1	x2	x3
Density	67.112	1.367	−0.8047
D	=		
	Speed		
Density	0		
K	=		
	Density		
x1	0.01385		
x2	0.03297		
x3	−0.03666		

Table 7
Quality metrics of the linear and neural state-space identified SISO models.

Linear and neural state-space models' quality metrics				
	SID	PEM	REG	NSS [64 64]
FPE	0.0033	0.0033	0.0033	0.0038
AIC	−5.1696e+04	−5.1841e+04	−5.1657e+04	−4.9150e+04
nAIC	−5.7131	−5.7211	−5.7109	−5.5715
BIC	−5.1696e+04	−5.1841e+04	−5.1657e+04	−4.9150e+04

$$
\begin{aligned}
&+ 0.01295z^{-4} + 0.05975z^{-5} + 0.05317z^{-6} \\
&+ 0.02052z^{-7} - 0.01284z^{-8} + 0.02178z^{-9} \\
&- 0.00194z^{-10} - 0.002855z^{-11} \\
B_{speed}(z) &= 0.1928 - 0.191z^{-1} - 0.08568z^{-2} + 0.08189z^{-3} \\
&+ 0.009049z^{-4} - 0.01923z^{-5} + 0.02256z^{-6} + 0.00423z^{-7} \\
&+ 0.0004569z^{-8} + 0.01832z^{-9} - 0.01252z^{-10} - 0.01541z^{-11} \\
&+ 0.002787z^{-12} - 0.001048z^{-13} - 0.006843z^{-14} - 0.000343z^{-15} + 0.01582z^{-16} - 0.008648z^{-17} .
\end{aligned}
$$

Again, for the NARX model, and using the same regressors as in the ARX model, we obtained the best results for two layers and 5 neurons per layer.

For illustration purposes, Fig. 9 shows the simulation and prediction responses of all the estimated models for a validation experiment. Fig. 10 presents the fitness boxplots of simulation and prediction responses of all the estimated SISO models. We observed no significant differences between the linear models. However, as with the multiple-input models, NARX was better for short-range predictions, while it deteriorated significantly for longer predictions.

Finally, Fig. 11 compares the performance of all the prediction models, MISO and SISO. Interestingly, for simulation and predictions above 15-steps ahead, the MISO models outperform single-input systems. We may conclude that modeling pedestrian inflow strongly influences prediction performance only for predicting density values above a specific threshold in terms of prediction range. For practical purposes, this threshold should be calculated to decide on using SISO or MISO modeling for each specific scenario. It is worth noting that if we model the pedestrian inflow, we must deploy the corresponding infrastructure to measure it in real-time.

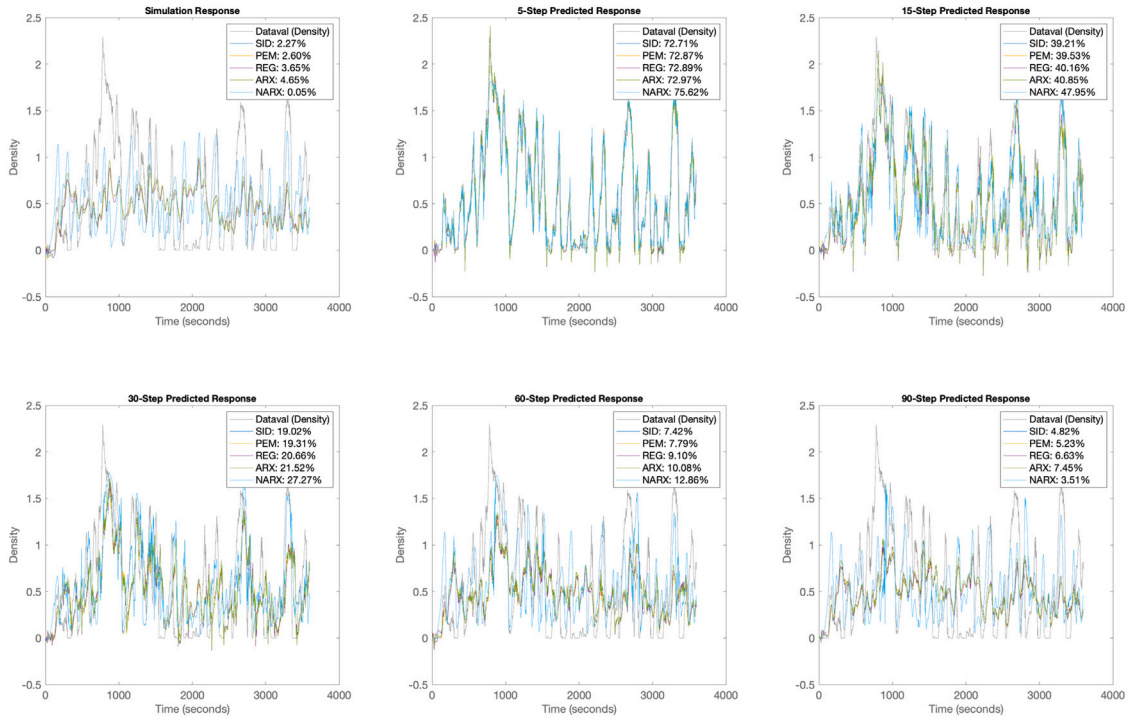

Fig. 9. Simulated and predicted response comparisons (Normalized Root Mean Square Error, NRMSE fitness) to validation data for SISO models. Dataval (Density) is the measured density. SID, PEM, and REG show the responses for the different linear state-space models: *subspace*, *prediction error minimization*, and *reduction of ARX model* methods. ARX and NARX show the responses for the linear and nonlinear input–output models.

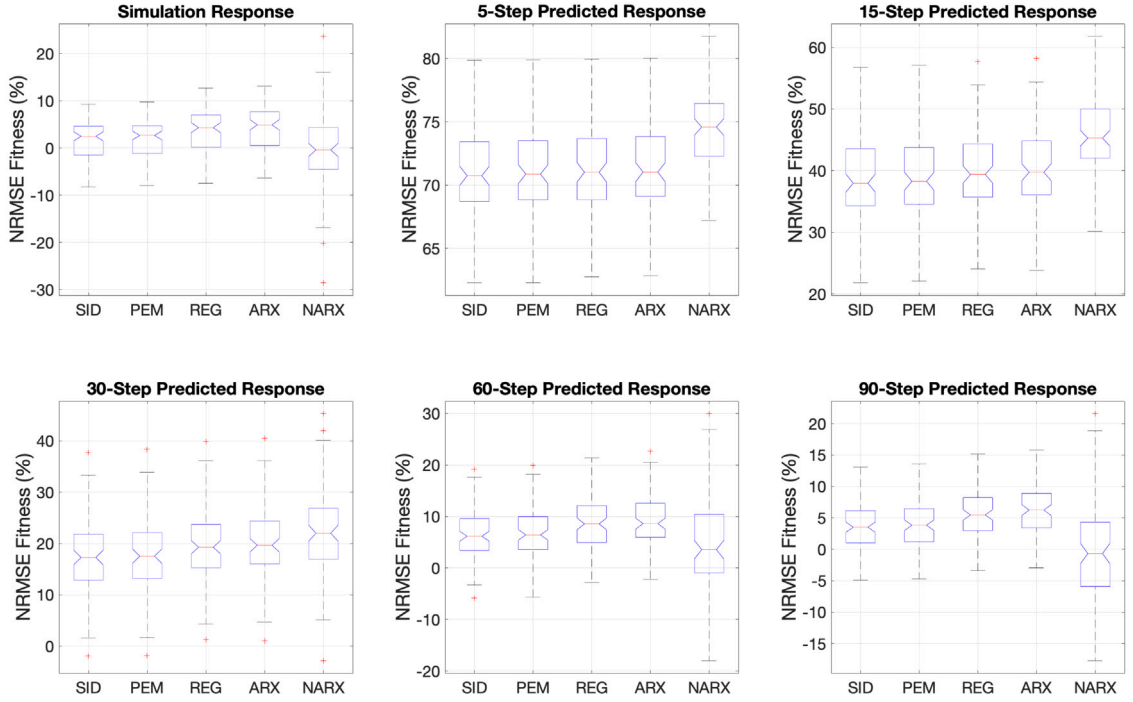

Fig. 10. NRMSE fitness boxplots of simulation and prediction responses of the SID, PEM, REG, ARX and NARX models for the SISO system.

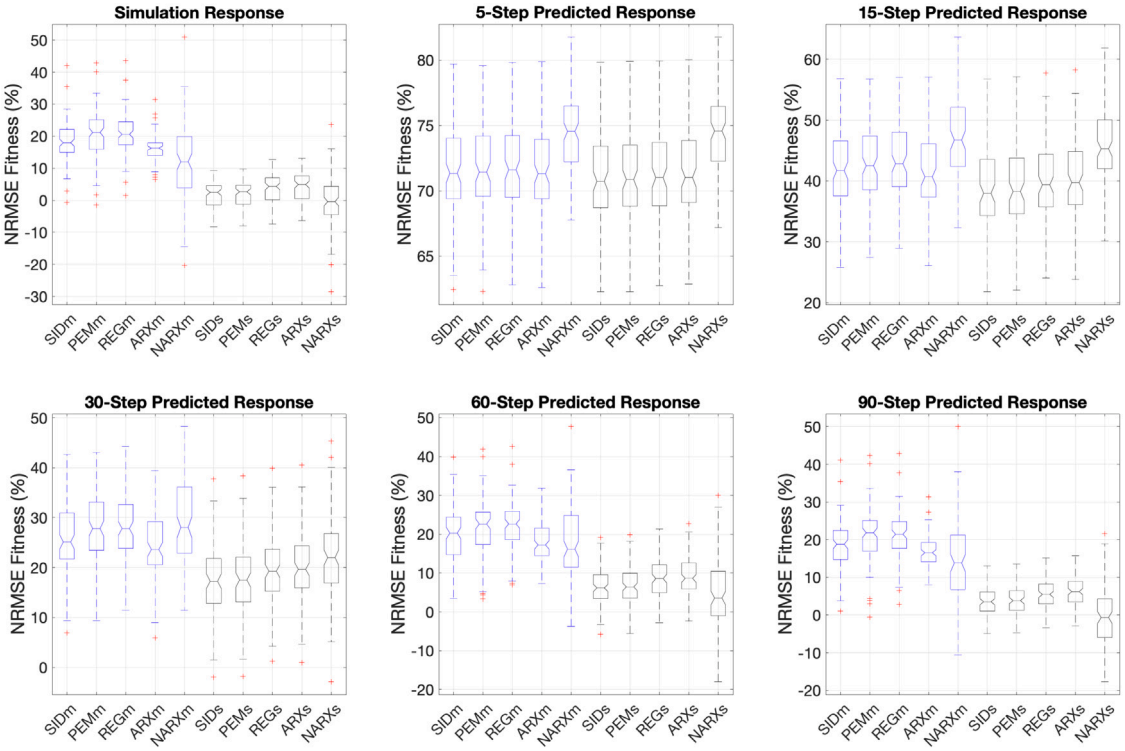

Fig. 11. NRMSE fitness boxplots of simulation and prediction responses of the SID, PEM, REG, ARX and NARX models for the MISO and SISO configurations. The boxplots in blue color represent the results for the MISO systems, while the black color shows the results for the SISO systems.

5. Discussion and conclusions

The application of the system identification methodology has proven its potential in building prediction models of crowd dynamics in the form of linear and nonlinear state-space and ARX models to be used as predictors in MPC controllers. The models have been estimated with input–output datasets obtained from microscopic simulation experiments using a general social force model. However, experiments with real crowds are also possible and should be considered for future research. Also, it is essential to clarify that future efforts in this research line should incorporate at least testing the obtained models with MPC controllers applied to real crowds.

In this work, we have taken a long corridor as an evacuation scenario because it is a ubiquitous structure. We are aware that it is not possible to generalize and extrapolate the conclusions drawn in this paper, regarding the relative performance of each model, to all possible evacuation scenarios with different layouts. However, applying the methodology of contrasting models to any scenario seems plausible, first testing linear ARX and state-space models, prioritizing the state-space models.

Regarding the microscopic simulation model used, it should be noted that there exist many variants of social force models requiring a different calibration process for different evacuation scenarios. Thus, for example, pedestrian interactions could be considered by including specific modeling of group behavior. Also, it would be possible to introduce complex models of response or behavior upon receipt of instructions or recommendations. Our strategy in this respect has focused the microscopic simulation on modeling forces and randomized perception. We found it reasonable to assume that in our controlled evacuation scenarios, at least for low-level emergencies, with pedestrians alerted of a speed control system, the influence of particular group behaviors or inaction upon receiving instructions is minimized. However, this does not preclude highlighting the importance of pedestrian adherence to the instructions issued by the control system (Feliciani et al., 2020). Especially in cases of emergency, it is necessary to investigate how to improve these levels of adherence, ensuring that individuals comply with the instructions provided. In this regard, neuroscience (Viloria et al., 2020) and social psychology (Moore et al., 2008) can contribute significantly to developing interventional, training, or informational mechanisms that result in the flow control systems' effective performance.

Finally, we want to highlight two relevant aspects of this research. Firstly, the good behavior of state-space models, with the advantages that this brings regarding stability and robustness testing and reduced order modeling. Secondly, the relative importance of explicit modeling of the input flow in constructing predictors.

Summarizing, the evidence from this research suggests the following:

- When comparing identification algorithms of linear state-space models for multiple-input-multiple-output prediction systems, including the subspace method, prediction error minimization, and reduction of ARX model, it has been observed that the best performance is obtained using the prediction error minimization algorithm, in terms of prediction error and model order.
- The neural state-space models improved for an increasing number of neurons and layers, though they did not outperform any of the linear state-space models. These results indicate that nonlinear modeling based on the system's internal state for our controlled evacuation scenario does not improve prediction performance.
- The linear state-space modeling outperforms ARX models. Interestingly, the linear state-space modeling is significantly better at increasing step-ahead predicted responses and at simulation, even though the order of the state-space model is 3 in contrast to the order 11 of the ARX model. It indicates that modeling the inner state of the evacuation process has a crucial and positive influence on modeling the system dynamics.
- The NARX models exhibit a better prediction fitness up to 15 steps ahead, worsening for increasing values. Interestingly, NARX has the best performance for short-term predictions compared to PEM and ARX, but PEM is preferred for long step-ahead predictions.
- In our study of how the extension of SISO modeling to MISO modeling influences the predictors' performance, we may conclude that modeling pedestrian inflow strongly influences prediction performance only from a prediction horizon value. For practical purposes, this threshold should be calculated to decide on using SISO or MISO modeling for each specific scenario. It is worth noting that if we model the pedestrian inflow, we must deploy the corresponding infrastructure to measure it in real-time.
- This study suggests that linear state-space models are the most suitable option in a corridor-type scenario and that including pedestrian inflow measurements significantly impact prediction performance.

These findings have important implications for real-world evacuation scenarios. Firstly, using the prediction error minimization algorithm to estimate linear state-space models, we enhance the accuracy of evacuation predictions, allowing for better decision-making in emergencies. Secondly, non-linear modeling based on the system's internal state may not be necessary or beneficial for controlled evacuation scenarios. Moreover, the superiority of linear state-space modeling over ARX models, despite the lower model order, highlights the significance of capturing the inner state of the evacuation process. On the other hand, the NARX models may be more appropriate depending on the prediction horizon needed when building the MPC controller.

The study emphasizes that modeling pedestrian inflow strongly influences prediction performance only from a prediction horizon value. This finding suggests that the choice between SISO and MISO modeling should be determined based on the specific scenario and the prediction horizon. If the MPC controller needs a relatively low prediction horizon, we may avoid measuring pedestrian inflow and minimize deployment complexity and cost.

As a logical extension to this research, we are investigating the design of MPC controllers that use the obtained predictors. Finally, we are exploring extending the single-output configuration to a multi-output configuration, including density measurements and control objectives in different areas of the evacuation scenario.

CRediT authorship contribution statement

Miguel A. Lopez-Carmona: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision. **Alvaro Paricio Garcia:** Software, Validation, Formal analysis, Investigation.

Acknowledgments

This work was partially supported by the Cátedra Masmovil for Advanced Network Engineering and Digital Services (MANEDS) CÁTEDRA2022-005UAH. The authors are most grateful to the reviewers for their comments and recommendations on the text.

References

Abdelghany, A., Abdelghany, K., Mahmassani, H., Alhalabi, W., 2014. Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities. European J. Oper. Res. 237, 1105–1118. http://dx.doi.org/10.1016/j.ejor.2014.02.054.

Akhter, F., Khadivizand, S., Siddiquei, H.R., Alahi, M.E.E., Mukhopadhyay, S., 2019. IoT enabled intelligent sensor node for smart city: Pedestrian counting and ambient monitoring. Sensors 19, 3374. http://dx.doi.org/10.3390/s19153374.

Ascione, F., Bianco, N., De Stasio, C., Mauro, G., Vanoli, G., 2015. Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. Energy Build. 111, http://dx.doi.org/10.1016/j.enbuild.2015.11.033.

Berceanu, C., Banu, I., Husebo, B.S., Patrascu, M., 2023. Predictive agent-based crowd model design using decentralized control systems. IEEE Syst. J. 17, 1383–1394. http://dx.doi.org/10.1109/JSYST.2022.3188339.

Bi, H., Gelenbe, E., 2019. A survey of algorithms and systems for evacuating people in confined spaces. Electronics 8, 711. http://dx.doi.org/10.3390/electronics8060711.

Bode, N.W.F., Kemloh Wagoum, A.U., Codling, E.A., 2015. Information use by humans during dynamic route choice in virtual crowd evacuations. R. Soc. Open Sci. 2, http://dx.doi.org/10.1098/rsos.140410.

Boomers, A.K., Boltes, M., Adrian, J., Beermann, M., Chraibi, M., Feldmann, S., Fiedrich, F., Frings, N., Graf, A., Kandler, A., Kilic, D., Konya, K., Küpper, M., Lotter, A., Lügering, H., Müller, F., Paetzke, S., Raytarowski, A.-K., Sablik, O., Schrödter, T., Seyfried, A., Sieben, A., Üsten, E., 2023. Pedestrian crowd management experiments: A data guidance paper. Collect. Dyn. 8, 1–57. http://dx.doi.org/10.17815/CD.2023.141.

Camacho, E.F., Alba, C.B., 2013. Model Predictive Control. Springer science & business media.

Chen, T., Ohlsson, H., Ljung, L., 2012. On the estimation of transfer functions, regularizations and Gaussian processes–revisited. Automatica 48, 1525–1535. http://dx.doi.org/10.1016/j.automatica.2012.05.026.

Chraibi, M., 2014. Oscillating behavior within the social force model.

Feliciani, C., Murakami, H., Shimura, K., Nishinari, K., 2020. Efficiently informing crowds – experiments and simulations on route choice and decision making in pedestrian crowds with wheelchair users. Transp. Res. C 114, 484–503. http://dx.doi.org/10.1016/j.trc.2020.02.019.
Feliciani, C., Nishinari, K., 2016. An improved cellular automata model to simulate the behavior of high density crowd and validation by experimental data. Physica A 451, 135–148. http://dx.doi.org/10.1016/j.physa.2016.01.057.
Feliciani, C., Nishinari, K., 2018. Measurement of congestion and intrinsic risk in pedestrian crowds. Transp. Res. C 91, 124–155. http://dx.doi.org/10.1016/j.trc.2018.03.027.
Gan, Q., Liu, Z., Liu, T., Chai, Y., 2022. An indoor evacuation guidance system with an AR virtual agent. Procedia Comput. Sci. 213, 636–642. http://dx.doi.org/10.1016/j.procs.2022.11.115.
Ghadami, A., Epureanu, B.I., 2022. Data-driven prediction in dynamical systems: Recent developments. Phil. Trans. R. Soc. A 380, 20210213. http://dx.doi.org/10.1098/rsta.2021.0213.
Gorbil, G., Gelenbe, E., 2011. Opportunistic communications for emergency support systems. Procedia Comput. Sci. 5, 39–47. http://dx.doi.org/10.1016/j.procs.2011.07.008.
Haghani, M., 2020. Optimising crowd evacuations: Mathematical, architectural and behavioural approaches. Saf. Sci. 128, 104745. http://dx.doi.org/10.1016/j.ssci.2020.104745.
Haghani, M., Ejtemai, O., Sarvi, M., Sobhani, A., Burd, M., Aghabayk, K., 2014. Random utility models of pedestrian crowd exit selection based on SP-off-RP experiments. Transp. Res. Procedia 2, 524–532. http://dx.doi.org/10.1016/j.trpro.2014.09.070.
Haghani, M., Lovreglio, Ruggiero, 2022. Data-based tools can prevent crowd crushes. Science 378, 1060–1061. http://dx.doi.org/10.1126/science.adf5949.
Haghani, M., Sarvi, M., 2017. Stated and revealed exit choices of pedestrian crowd evacuees. Transp. Res. B 95, 238–259.
Haghani, M., Sarvi, M., 2019a. Imitative (herd) behaviour in direction decision-making hinders efficiency of crowd evacuation processes. Saf. Sci. 114, 49–60. http://dx.doi.org/10.1016/j.ssci.2018.12.026.
Haghani, M., Sarvi, M., 2019b. Simulating dynamics of adaptive exit-choice changing in crowd evacuations: Model implementation and behavioural interpretations. Transp. Res. C 103, 56–82. http://dx.doi.org/10.1016/j.trc.2019.04.009.
Helbing, D., Buzna, L., Johansson, A., Werner, T., 2005. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. Transp. Sci. 39, 1–24.
Helbing, D., Farkas, I., Vicsek, T., 2000. Simulating dynamical features of escape panic. Nature 407, 487–490. http://dx.doi.org/10.1038/35035023.
Helbing, D., Molnár, P., 1995. Social force model for pedestrian dynamics. Phys. Rev. E 51, 4282–4286. http://dx.doi.org/10.1103/PhysRevE.51.4282.
Helbing, D., Mukerji, P., 2012. Crowd disasters as systemic failures: Analysis of the love parade disaster. EPJ Data Sci. 1, 1–40.
Jansson, M., 2003. Subspace identification and ARX modeling. IFAC Proc. Vol. 36, 1585–1590. http://dx.doi.org/10.1016/S1474-6670(17)34986-8.
Johansson, F., Duives, D., Daamen, W., Hoogendoorn, S., 2014. The many roles of the relaxation time parameter in force based models of pedestrian dynamics. Transp. Res. Procedia 2, 300–308. http://dx.doi.org/10.1016/j.trpro.2014.09.057.
Lakoba, T.I., Kaup, D.J., Finkelstein, N.M., 2005. Modifications of the Helbing-Molnár-Farkas-Vicsek social force model for pedestrian evolution. SIMULATION 81, 339–352. http://dx.doi.org/10.1177/0037549705052772.
Larimore, W., 1990. Canonical variate analysis in identification, filtering, and adaptive control. In: 29th IEEE Conference on Decision and Control, Vol. 2. pp. 596–604. http://dx.doi.org/10.1109/CDC.1990.203665.
Lewis, F.L., Vrabie, D., Vamvoudakis, K.G., 2012. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. IEEE Control Syst. Mag. 32, 76–105. http://dx.doi.org/10.1109/MCS.2012.2214134.
Li, Y., Chen, M., Dou, Z., Zheng, X., Cheng, Y., Mebarki, A., 2019. A review of cellular automata models for crowd evacuation. Phys. A 526, 120752. http://dx.doi.org/10.1016/j.physa.2019.03.117.
Li, D., Ranjitkar, P., 2015. A fuzzy logic-based variable speed limit controller. J. Adv. Transp. 49, 913–927. http://dx.doi.org/10.1002/atr.1320.
Liao, C., Guo, H., Zhu, K., Shang, J., 2019. Enhancing emergency pedestrian safety through flow rate design: Bayesian-Nash equilibrium in multi-agent system. Comput. Ind. Eng. 137, 106058. http://dx.doi.org/10.1016/j.cie.2019.106058.
Liu, H., Lu, D., Zhang, G., Hong, X., Liu, H., 2021. Recurrent emotional contagion for the crowd evacuation of a cyber–physical society. Inform. Sci. 575, 155–172. http://dx.doi.org/10.1016/j.ins.2021.06.036.
Liu, X., Qiu, L., Rodriguez, J., Wu, W., Ma, J., Peng, Z., Wang, D., Fang, Y., 2022. Data- driven neural predictors-based robust MPC for power converters. IEEE Trans. Power Electron. 37, 11650–11661. http://dx.doi.org/10.1109/TPEL.2022.3171100.
Ljung, L., 1999. System Identification: Theory for the User, second ed. Prentice-Hall PTR, NJ.
Ljung, L., Chen, T., Mu, B., 2020. A shift in paradigm for system identification. Internat. J. Control 93, 173–180. http://dx.doi.org/10.1080/00207179.2019.1578407.
Lopez-Carmona, M.A., 2022. System identification for the design of behavioral controllers in crowd evacuations. Transp. Res. C 144, 103913. http://dx.doi.org/10.1016/j.trc.2022.103913.
Lopez-Carmona, M.A., Paricio-Garcia, A., 2020. LED wristbands for cell-based crowd evacuation: An adaptive exit-choice guidance system architecture. Sensors 20 (6038), http://dx.doi.org/10.3390/s20216038.
Lopez-Carmona, M.A., Paricio Garcia, A., 2021. CellEVAC: An adaptive guidance system for crowd evacuation through behavioral optimization. Saf. Sci. 139, 105215. http://dx.doi.org/10.1016/j.ssci.2021.105215.
Lopez-Carmona, M.A., Paricio Garcia, A., 2022. Adaptive cell-based evacuation systems for leader-follower crowd evacuation. Transp. Res. C 140, 103699. http://dx.doi.org/10.1016/j.trc.2022.103699.
Lovreglio, R., Fonzone, A., dell'Olio, L., Borri, D., 2016. A study of herding behaviour in exit choice during emergencies based on random utility theory. Saf. Sci. 82, 421–431. http://dx.doi.org/10.1016/j.ssci.2015.10.015.
Lovreglio, R., Ronchi, E., Kinsey, M.J., 2020. An online survey of pedestrian evacuation model usage and users. Fire Technol. 56, 1133–1153. http://dx.doi.org/10.1007/s10694-019-00923-8.
Mohd Ibrahim, A., Venkat, I., Wilde, P.D., 2017. Uncertainty in a spatial evacuation model. Physica A 479, 485–497. http://dx.doi.org/10.1016/j.physa.2017.03.024.
Moore, S.C., Flajšlik, M., Rosin, P.L., Marshall, D., 2008. A particle model of crowd behavior: Exploring the relationship between alcohol, crowd dynamics and violence. Aggress. Violent Behav. 13, 413–422. http://dx.doi.org/10.1016/j.avb.2008.06.004.
Murakami, H., Feliciani, C., Shimura, K., Nishinari, K., 2020. A system for efficient egress scheduling during mass events and small-scale experimental demonstration. R. Soc. Open Sci. 7, 201465. http://dx.doi.org/10.1098/rsos.201465.
Ogata, K., 2010. Modern Control Engineering. Prentice Hall.
Pan, X., Han, C.S., Dauber, K., Law, K.H., 2007. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. AI Soc. 22, 113–132. http://dx.doi.org/10.1007/s00146-007-0126-1.
Pelechano, N., Malkawi, A., 2008. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. Autom. Constr. 17, 377–385. http://dx.doi.org/10.1016/j.autcon.2007.06.005.
Re3data.Org, 2020. Pedestrian dynamics data archive. http://dx.doi.org/10.17616/R31NJMT6.
Schwenzer, M., Ay, M., Bergs, T., Abel, D., 2021. Review on model predictive control: An engineering perspective. Int. J. Adv. Manuf. Technol. 117, 1327–1349. http://dx.doi.org/10.1007/s00170-021-07682-3.

Singh, R., Sznaier, M., 2021. On identification of nonlinear ARX models with sparsity in regressors and basis functions. IFAC-PapersOnLine 54, 720–725. http://dx.doi.org/10.1016/j.ifacol.2021.08.446.

Sirmatel, I.I., Geroliminis, N., 2018. Mixed logical dynamical modeling and hybrid model predictive control of public transport operations. Transp. Res. B 114, 325–345. http://dx.doi.org/10.1016/j.trb.2018.06.009.

Steffen, B., Seyfried, A., 2010. Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. Phys. A 389, 1902–1910.

Train, K., 2009. Discrete Choice Methods with Simulation. Cambridge University Press.

van der Wal, C.N., Formolo, D., Robinson, M.A., Minkov, M., Bosse, T., 2017. Simulating crowd evacuation with socio-cultural, cognitive, and emotional elements. In: Mercik, J., Nguyen, N.T., Kowalczyk, R. (Eds.), Transactions on Computational Collective Intelligence. In: Lecture Notes in Computer Science, vol. XXVII, Springer International Publishing, Cham, pp. 139–177.

van der Wal, C.N., Robinson, M.A., Bruine de Bruin, W., Gwynne, S., 2021. Evacuation behaviors and emergency communications: An analysis of real-world incident videos. Saf. Sci. 136, 105121. http://dx.doi.org/10.1016/j.ssci.2020.105121.

van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, H.J., 2002. Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. Transp. Res. Rec. 1811, 30–39. http://dx.doi.org/10.3141/1811-04.

Van Overschee, P., De Moor, B., 1996. Subspace Identification for Linear Systems. Springer US, Boston, MA, http://dx.doi.org/10.1007/978-1-4613-0465-4.

Verhaegen, M., 1994. Identification of the deterministic part of MIMO state space models given in innovations form from input–output data. Automatica 30, 61–74. http://dx.doi.org/10.1016/0005-1098(94)90229-1.

Viloria, A., Pineda Lezama, O.B., Vargas, J., 2020. Analysis of crowd behavior through pattern virtualization. Procedia Comput. Sci. 175, 102–107. http://dx.doi.org/10.1016/j.procs.2020.07.017.

Wang, H.-R., Chen, Q.-G., Yan, J.-B., Yuan, Z., Liang, D., 2015. Emergency guidance evacuation in fire scene based on pathfinder. In: Proceedings - 7th International Conference on Intelligent Computation Technology and Automation, ICICTA 2014. pp. 226–230. http://dx.doi.org/10.1109/ICICTA.2014.62.

Yao, Z., Zhang, G., Lu, D., Liu, H., 2020. Learning crowd behavior from real data: A residual network method for crowd simulation. Neurocomputing 404, 173–185. http://dx.doi.org/10.1016/j.neucom.2020.04.141.

Zhang, Z., Jia, L., 2021. Optimal guidance strategy for crowd evacuation with multiple exits: A hybrid multiscale modeling approach. Appl. Math. Model. 90, 488–504. http://dx.doi.org/10.1016/j.apm.2020.08.075.

Zhang, Z., Jia, L., Qin, Y., 2017. Optimal number and location planning of evacuation signage in public space. Saf. Sci. 91, 132–147. http://dx.doi.org/10.1016/j.ssci.2016.07.021.

Zhang, J., Seyfried, A., 2013. Empirical characteristics of different types of pedestrian streams. Procedia Eng. 62, 655–662. http://dx.doi.org/10.1016/j.proeng.2013.08.111.

Zia, K., Ferscha, A., 2020. An agent-based model of crowd evacuation: Combining individual, social and technological aspects. In: Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation SIGSIM-PADS '20. Association for Computing Machinery, New York, NY, USA, pp. 129–140. http://dx.doi.org/10.1145/3384441.3395973.