POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Neuro-Inspired Control Architecture to Enhance Robot Self-Preservation and Adaptation in Autonomous Navigation Tasks

Original
A Neuro-Inspired Control Architecture to Enhance Robot Self-Preservation and Adaptation in Autonomous Navigation Tasks / Usai, A., Rizzo, A.. - In: IEEE ROBOTICS AND AUTOMATION LETTERS. - ISSN 2377-3766. - ELETTRONICO. - 10:8(2025), pp. 8491-8497. [10.1109/lra.2025.3583630]

Availability:
This version is available at: 11583/3001941 since: 2025-07-28T08:43:35Z

Publisher:
IEEE

Published
DOI:10.1109/lra.2025.3583630

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

A Neuro-Inspired Control Architecture to Enhance Robot Self-Preservation and Adaptation in Autonomous Navigation Tasks

Andrea Usai and Alessandro Rizzo

***Abstract*—Ensuring survival and self-preservation is essential to design intelligent robots that adapt to dynamic and unfamiliar environments. Inspired by the dual-pathway model from neuroscience, we introduce a control architecture designed to ensure the adaptability of robotic behavior during navigation. This approach parallels the neuroscientific "Low Road" paradigm by incorporating constructs resembling the thalamus, implemented as a nonlinear filter; the amygdala, modeled as a Soft Actor-Critic (SAC) reinforcement learning agent; and the brainstem-cerebellum connection, represented by a Nonlinear Model Predictive Controller (NMPC). Our findings indicate superior adaptiveness, generalizability, and computational efficiency compared to standard NMPCs and Artificial Potential Fields in both static and dynamic environments with obstacles of varying risk levels.**

***Index Terms*—Bioinspired robot learning, neurorobotics, cognitive control architectures.**

I. Introduction

THE use of autonomous robots, such as rovers and drones, is becoming increasingly common in a wide range of applications, including infrastructure inspections [1], search and rescue missions [2] or social navigation [3]. In these contexts, autonomous systems must not only fulfill their operational tasks, but also prioritize their meta-goal of survival and self-preservation, especially in complex and unpredictable environments [4]. Thus, ensuring adaptive navigation is crucial for long-term system safety and functionality. Typical navigation methods can be categorized into model-based and learning-based approaches. Model-based methods, such as the well-known Artificial Potential Field (APF) [5] and Optimal Reciprocal Collision Avoidance (ORCA) [6], compute control inputs using physical or kinematic models whose terms are described by tunable parameters or weights. Recently, Nonlinear Model Predictive Controllers (NMPCs) have gained traction in autonomous driving [7] and robotic applications [8], [9] due to their ability to compute the optimal control input accounting for the system dynamics and operational constraints. Similarly to other methods, the effectiveness of NMPC is highly dependent on weight tuning, which is often performed manually and tailored to specific scenarios. Such a tuning can be particularly challenging and time-consuming for complex and dynamic environments, thus limiting their generalizability to diverse contexts [10]. In contrast, learning-based methods, primarily using Reinforcement Learning (RL) [11], [12], [13], learn navigation policies directly through interaction with the environment. However, training RL in large and non-stationary environments remains a challenge due to increased variability, resulting in inefficient learning [13]. To mitigate these issues, some efforts reduce the random distribution of training data, resulting in overfitting and poor generalizability to unseen environments [11], [12]. In general, both model-based and learning-based approaches have significant limitations in terms of adaptability and generalization.

The integration of artificial emotions within robotic systems presents a promising avenue for addressing these challenges [4]. Emotions, as adaptive evolutionary mechanisms, are integral in shaping behavior by guiding decision-making processes in response to perceived threats or opportunities [14]. Implementing these mechanisms within robotic architectures confers substantial benefits. As demonstrated in [4] and [15], the introduction of the concept of risk-to-self and the fusion of emotions with cognitive processes within a robot's information processing framework are imperative for enhancing truly autonomous and intelligent behaviors. In this regard, employing neuro-inspired, emotion-based control architectures allows robots to achieve heightened adaptability in unforeseen circumstances, thereby ensuring improved performance in dynamic and complex environments [16].

Among the vast atlas of emotions, fear is the most straightforward to self-preservation and adaptation [17]. From a neurological point of view, the *dual-pathway hypothesis*, proposed by LeDoux [18], explains how the brain processes fear stimuli and influences behaviors (Fig. 1(a)). In this framework, two neural pathways involving the amygdala are considered: the *Low Road* and the *High Road*. The Low Road is a direct thalamus-amygdala pathway that triggers rapid and unconscious responses to potentially dangerous stimuli from the thalamus, provid-

Received 3 April 2025; accepted 10 June 2025. Date of publication 27 June 2025; date of current version 14 July 2025. This article was recommended for publication by Associate Editor L. Jamone and Editor T. Ogata upon evaluation of the reviewers' comments. This work was supported by MOST – Sustainable Mobility National Research Center funded by European Union Next-GenerationEU (PNRR – Missione 4, Componente 2, Investimento 1.4 – D.D. 1033 17/06/2022, CN00000023) and in part by FAIR - Future Artificial Intelligence Research funded by European Union Next-GenerationEU (PNRR – Missione 4, Componente 2, Investimento 1.3 – D.D. 1555 11/10/2022, PE00000013). *(Corresponding author: Alessandro Rizzo.)*

The authors are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10126 Torino, Italy (e-mail: andrea.usai@polito.it; alessandro.rizzo@polito.it).

Digital Object Identifier 10.1109/LRA.2025.3583630

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

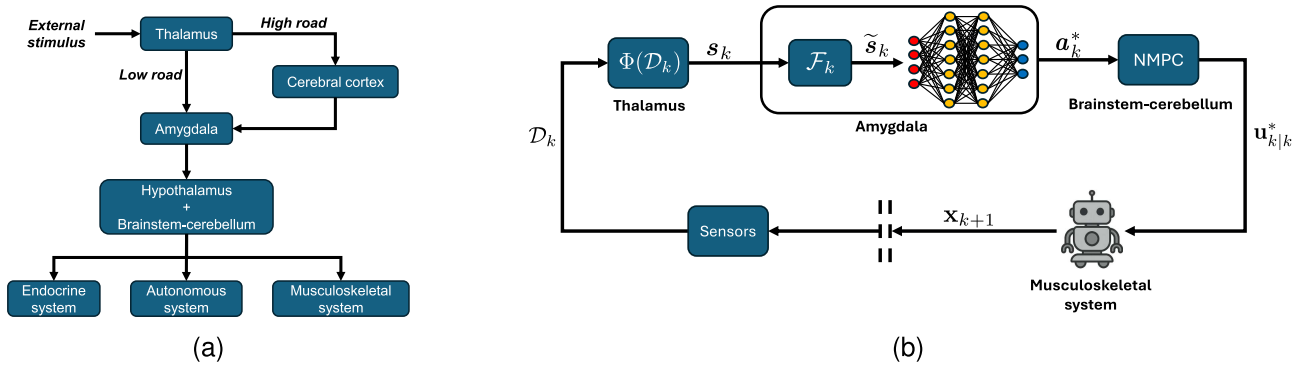

Fig. 1. (a) Schematic representation of the dual-pathway model. - (b) Control architecture based on the Low Road paradigm.

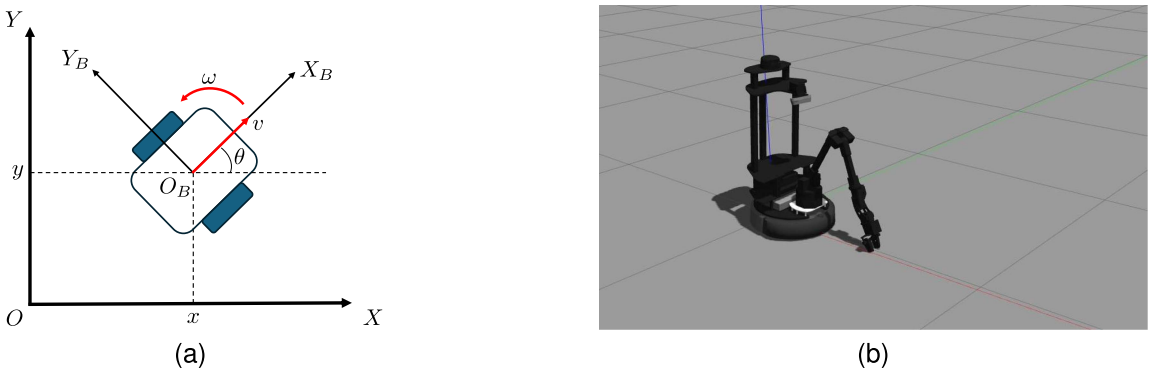

Fig. 2. (a) Coordinate systems of the wheeled mobile robot: $[x_k, y_k, \theta_k]$ are the position and the orientation of the robot in the inertial frame $\{O, X, Y\}$, v_k and ω_k are the linear and angular velocity of the robot in the body frame $\{O_B, X_B, Y_B\}$, respectively. - (b) Locobot WX250s used for Gazebo simulations.

ing an essential survival mechanism [19]. Here, the amygdala assigns emotional valence to threats using a punishment-reward mechanism [20]. If a threat is detected, the amygdala signals the brainstem and hypothalamus to initiate a physiological response by activating the endocrine, autonomic, and musculoskeletal systems. The latter is coordinated by the cerebellum through its connection to the brainstem. In contrast, the High Road is a slower indirect pathway through the cerebral cortex, allowing for a more precise evaluation to modulate the initial response elicited by the Low Road. Based on these neuroscientific evidence, in this letter we parallel the core functionalities of the Low Road's neural subsystems to design a neuro-inspired control system that ensures safe robot navigation by accounting for self-preservation in environments with varying obstacle danger levels (Fig. 2(b)). In our framework, the amygdala is modeled as a Soft Actor-Critic (SAC) reinforcement learning agent [21], able to assess the risk of the situation and organize a coherent motor response by means of the brainstem-cerebellum connection. The latter, modeled as a NMPC, receives the response of the amygdala, which corresponds to a suitable set of weights for the NMPC optimization problem. Then, it uses such weights to compute the optimal control input for the robot. Such a dynamic tuning allows the amygdala to modulate the robot's behavior based on the current scenario. Although most of the existing adaptive NMPC schemes in robotics focus on improving the tracking of the reference path [10] or compensating model uncertainties [22], to the best of our knowledge, this is the first attempt to develop an adaptive reinforcement learning-based mechanism with neuroscientific plausibility to enhance robot self-preservation. The main contributions of this letter are:

- a neuro-inspired control architecture for robot navigation that prioritizes self-preservation, based on the perceived level of danger;
- a novel reinforcement learning-based mechanism for tuning NMPC weights, particularly challenging in dynamic environments;
- a training approach that leverages NMPC constraints, improving robustness and generalizability.

Our results show superior safety, generalizability and computational efficiency compared to state-of-the-art navigation methods in both static and dynamic environments. The remainder of the letter is organized as follows. Section II provides a detailed description of the proposed architecture. Section III outlines the training procedure for the amygdala and the experimental setup. Section IV presents and analyzes experimental results. Finally, Section V concludes the letter and discusses future work.

II. Low-Road Architecture

Figure 1(b) illustrates the proposed architecture for the Low-Road, which is composed of three subsystems, named after their neurological counterpart:

1) Thalamus: The thalamus is responsible for extracting sensory information from incoming stimuli and transmitting it to

the amygdala for risk assessment. We represent this working principle through a generic nonlinear filter function $\Phi(\cdot)$, which extracts the robot's interoceptive and exteroceptive information from a set of raw sensory data $\mathcal{D}_k$ collected at discrete time k from sensors available on the considered robotic platform (e.g. cameras, lidars, UWB, GPS, etc...). Such a filter generates an observation vector

$$\boldsymbol{s}_k = \Phi(\mathcal{D}_k) = [\varphi_k^{(1)}, \varphi_k^{(2)}, \ldots, \varphi_k^{(p)}]^T \tag{1}$$

composed of $p \in \mathbb{N}$ main *features* $\varphi_k^{(\cdot)}$ that describe the situation that the robot is facing at each discrete time instant k. Here, we consider as features all obstacles' positions in the environment $\mathbf{x}_k^{obs_j}$ with $j = 1, 2, \ldots, n$, and n the total number of obstacles, the past control input $\mathbf{u}_{k-1}^*$, the robot current state $\mathbf{x}_k$ (position and orientation) and the goal position $\mathbf{x}_r$ that the robot has to reach, i.e. the reference state of the control algorithm, yielding

$$\boldsymbol{s}_k = \left[\mathbf{x}_k^{obs_1}, \ldots, \mathbf{x}_k^{obs_n}, \mathbf{u}_{k-1}^*, \mathbf{x}_k, \mathbf{x}_r\right]^T. \tag{2}$$

2) Amygdala: The main role of the amygdala is to quantify the dangerousness of the perceived environment and to produce a coherent response represented by the SAC agent's output. Since the introduction of LeDoux's dual-pathway model aimed at explicating the physiological response mechanisms by which the body responds to fear, we adopt the term "fear level" as a broad and pragmatic descriptor that encapsulates the comprehensive perceived threat quantified by the amygdala, which is instrumental in informing its decision-making processes. Considering the reactive nature and rapid responsiveness of the "Low Road" mechanism, it is postulated that such "fear level" can be derived from readily quantifiable variables, directly pertinent to the task at hand. Given the preliminary nature of this study, a navigation task was selected for our analysis. During the initial processing stages, the amygdala processes the feature vector $\mathbf{s}_k$ originating from the thalamus to extract a *relative observation vector* $\bar{\mathbf{s}}_k$ defined as

$$\bar{\mathbf{s}}_k = \left[\frac{1}{d_k^{obs_1}}, \ldots, \frac{1}{d_k^{obs_n}}\right]^T, \tag{3}$$

where $d_k^{obs_j}$ represents the relative distance between the robot and the j-th obstacle at time instant k. The vector $\bar{\mathbf{s}}_k$ is then used to compute the current fear level $\mathcal{F}_k$ as

$$\mathcal{F}_k = \sigma(w_0 + \boldsymbol{w} \cdot \bar{\mathbf{s}}_k) \in [0, 1], \tag{4}$$

where $\boldsymbol{w} = [w_{obs_1}, \ldots, w_{obs_n}]$ denotes a weight vector that modulates the impact of each feature in $\bar{\mathbf{s}}_k$ on the fear level, while w_0 serves as a constant bias refining the configuration of the fear level function. These parameters can be interpreted as a way of capturing obstacle characteristics that influence fear level but are not directly related to the task – such as social rules related to the context or safety constraints – whose quantification is challenging within the sole reactive paradigm of the Low Road. Concurrently, $\sigma(\cdot)$ denotes a sigmoid function, which normalizes the fear level within $[0, 1]$. This normalization produces a uniform observation state variable, thereby augmenting learning stability and enabling the formulation of a more effective reward function (see Section III). The resulting fear level $\mathcal{F}_k$ is then utilized to build an *augmented observation vector* $\widetilde{\mathbf{s}}_k$ defined as

$$\widetilde{\boldsymbol{s}}_k = [\boldsymbol{s}_k, \mathcal{F}_k]^T = \left[\mathbf{x}_k^{obs_1}, \ldots, \mathbf{x}_k^{obs_n}, \mathbf{u}_{k-1}^*, \mathbf{x}_k, \mathbf{x}_r, \mathcal{F}_k\right]^T. \tag{5}$$

The latter is fed as input to the SAC agent to compute the action $\boldsymbol{a}_k = [\text{diag}(\mathbf{Q}_k), \text{diag}(\mathbf{R}_k), \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k] \in \mathbb{R}^8$, which consists of the weights for the NMPC optimization problem. The SAC agent uses a stochastic Gaussian policy with outputs bounded in [0,1], hence, the resulting action $\boldsymbol{a}_k$ is denormalized through an element-wise multiplication

$$\boldsymbol{a}_k^* = \boldsymbol{a}_k \odot \boldsymbol{a}_{max}, \tag{6}$$

where $\boldsymbol{a}_k^* = \left[\text{diag}(\mathbf{Q}^*), \text{diag}(\mathbf{R}^*), \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*\right] \in \mathbb{R}^8$ is the denormalized action and $\boldsymbol{a}_{\max} = \left[\text{diag}(\mathbf{Q}_{\max}), \text{diag}(\mathbf{R}_{\max}), \boldsymbol{\alpha}_{\max}, \boldsymbol{\beta}_{\max}\right] \in \mathbb{R}^8$ is a vector containing the maximum allowable values for the corresponding NMPC weights.

3) Brainstem-Cerebellum Connection: The brainstem-cerebellum connection is essential for fine motor control and movement coordination through sensory feedback, ensuring precise adjustments and smooth execution under constraints. We parallel this functionality using an NMPC, which optimizes control inputs based on the system's dynamics and constraints, similar to the cerebellum's role. To exemplify our strategy, we consider a wheeled mobile robot modeled as a unicycle (Fig. 2(a)), whose discrete-time kinematics can be expressed as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + T_s \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = f_d(\mathbf{x}_k, \mathbf{u}_k), \tag{7}$$

where $\mathbf{x}_k = [x_k \quad y_k \quad \theta_k]^T$ and $\mathbf{u}_k = [v_k \quad \omega_k]^T$ are the state vector and the control input vector at the discrete time instant k, respectively, and T_s is the sampling time. As previously mentioned, at each time step k, the amygdala provides weights matrices $\mathbf{Q}_k^*$, $\mathbf{R}_k^*$, $\boldsymbol{\alpha}_k^*$ and $\boldsymbol{\beta}*_k$ to the NMPC, which computes the optimal control input sequence $\mathbf{u}^*(\cdot|k)$ over the prediction horizon $N \in \mathbb{N}$ by solving the following optimal control problem:

$$\begin{aligned}
\min_{\mathbf{u}(\cdot|k)} \; & \sum_{i=0}^{N-1} \ell(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}, \mathbf{Q}_k^*, \mathbf{R}_k^*) \\
& + B(\mathbf{x}_{k+i|k}, \mathbf{x}_{k+i|k}^{obs_j}, \boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*) \\
\text{s.t.} \quad & \mathbf{x}_{k|k} = \mathbf{x}_k, \\
& \mathbf{x}_{k+i+1|k} = f_d(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}), \quad \forall i = 0, \ldots, N-1, \\
& \mathbf{u}_{k+i|k} \in \mathbb{U}, \quad \mathbf{x}_{k+i+1|k} \in \mathbb{X}, \quad \forall i = 0, \ldots, N-1, \\
& \mathbf{x}_{k+i+1|k}^{obs_j} = \mathbf{x}_{k+i|k}^{obs_j} + T_s \dot{\mathbf{x}}_k^{obs_j}, \quad j = 1, \ldots, n,
\end{aligned} \tag{8}$$

where $\mathbb{U}$ is the set of feasible control inputs and $\mathbb{X}$ is the set of feasible states. The stage cost $\ell(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}, \mathbf{Q}_k^*, \mathbf{R}_k^*) = \|\mathbf{x}_r - C\mathbf{x}_{k+i|k}\|_{\mathbf{Q}_k^*}^2 + \|\mathbf{u}_{k+i|k}\|_{\mathbf{R}_k^*}^2$ is intended to guide the robot toward the goal $\mathbf{x}_r$, while the term $B(\mathbf{x}_{k+i|k}, \boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*) = \sum_{j=1}^n -\alpha_k^{obs_j} \log(\beta_k^{obs_j} \|\mathbf{x}_{k+i|k}^{obs_j} - C\mathbf{x}_{k+i|k}\|^2)$ is a logarithmic

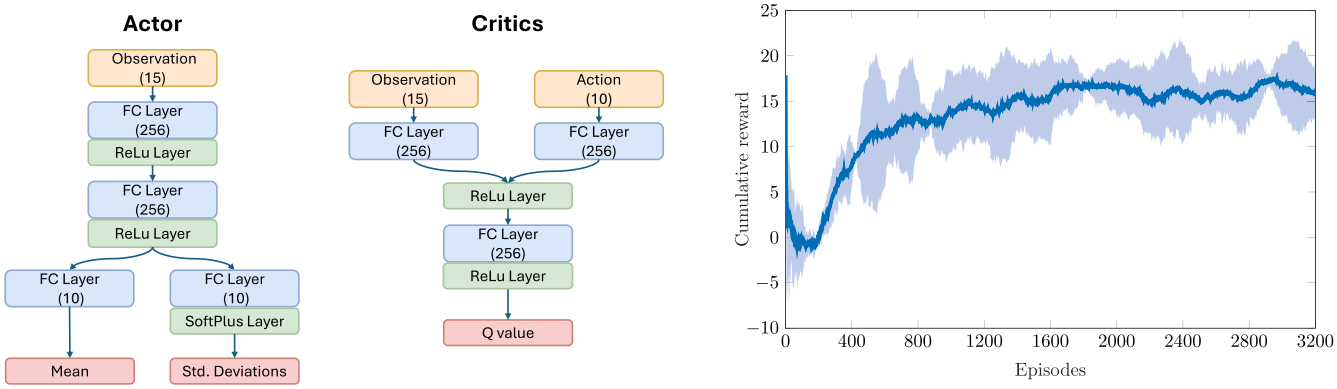

Fig. 3. (a) Structure of the actor and critic networks in the SAC agent, characterized by the following hyperparameters [21]: Discount factor = 0.99, Replay buffer size = 10^4, Mini batch size = 64, Target smooth factor = 10^{-4}, Actor and critics learning rates = 10^{-3}, Entropy weight = 1, Entropy learning rate = $3 \cdot 10^{-4}$. - (b) Average cumulative reward of the implemented SAC agent. The marked line is the mean value of the cumulative reward across five seeds; the lighter shaded area is the ± 3 standard deviation confidence interval.

penalty function designed to move the robot away from the j-th harmful environmental obstacle in position $\mathbf{x}_{k+i|k}^{obs_j}$. To ensure a better adaptive behavior, the amygdala is able to estimate different parameters $\alpha_k^{obs_j}$ and $\beta_k^{obs_j}$ according to the dangerousness of the j-th obstacles. Finally, the term $C = \text{diag}([1, 1, 0])$ is a matrix that extracts the robot's position from the state vector $\mathbf{x}_{k+i|k}$.

III. Implementation

A. Training of the Amygdala

The structure of the actor and critics, as well as the corresponding hyperparameters used in the SAC algorithm, are illustrated in Fig. 3(a). Here and henceforth, unless otherwise specified, the parameter values have been set empirically. During the training phase, the selection of the action of the SAC agent is guided by a reward function defined as

$$R(\tilde{\boldsymbol{s}}_k, \boldsymbol{a}_k) = R_F + R_I + R_R, \tag{9}$$

whose components are defined as

$$R_F = \begin{cases} -20 & \text{if collision,} \\ -2.5\mathcal{F}_k & \text{otherwise;} \end{cases}$$

$$R_I = \begin{cases} -20 & \text{if infeasibility,} \\ 0 & \text{otherwise;} \end{cases}$$

$$R_R = \begin{cases} 10 & \text{if goal reached,} \\ 2(d_{k-1}^{ref} - d_k^{ref}) & \text{otherwise.} \end{cases}$$

Here, R_F penalizes states that induce an excessive fear level or lead to collisions with obstacles, R_I penalizes actions that result in infeasible solutions for the NMPC, while R_R encourages the selection of weights that help the robot to approach the target position by reducing the distance $d_k^{ref} = \|\mathbf{x}_r - C\mathbf{x}_k\|$ at time instant k with respect to the previous time instant $k-1$.

We preliminary implemented the proposed architecture in MATLAB/Simulink and trained the amygdala in a static environment with two types of obstacles: dangerous and non-dangerous. We set $w_0 = -5$, while the weights w_{obs_j} are equal to 3 for dangerous obstacles and 1.5 otherwise. To improve the generalizability of the training process, at the beginning of each new episode ($k = 0$), the environment reinitializes the robot's initial state $\mathbf{x}_0$, the obstacles initial positions $\mathbf{x}_0^{obs_j}$ and the goal position $\mathbf{x}_r$ by sampling from uniform random distributions

$$\begin{aligned} \mathbf{x}_0 &= \begin{bmatrix} x_0, y_0, \theta_0 \end{bmatrix}^T \sim \begin{bmatrix} \mathcal{U}(0, 0.5), \mathcal{U}(0, 0.5), \mathcal{U}(0, \pi/2) \end{bmatrix}^T, \\ \mathbf{x}_0^{obs_j} &= \begin{bmatrix} x_0^{obs_j}, y_0^{obs_j} \end{bmatrix}^T \sim \begin{bmatrix} \mathcal{U}(0, 6), \mathcal{U}(1.5, 5) \end{bmatrix}^T, \\ \mathbf{x}_r &= \begin{bmatrix} x_r, y_r \end{bmatrix}^T \sim \begin{bmatrix} \mathcal{U}(5.5, 6.5), \mathcal{U}(5.5, 6.5) \end{bmatrix}^T, \end{aligned} \tag{10}$$

where $\mathcal{U}(a, b)$ represents a uniform random distribution over the interval $[a, b]$. This randomization strategy offers a more diverse sampling space than other state-of-the-art methods [11] allowing for a greater generalization during training. We trained the SAC agent over 3200 episodes, using five different seeds to demonstrate the robustness and stability of the training process. For each seed, we computed the cumulative reward as the moving average of the total rewards from the last 150 episodes. The results, averaged across the five different seeds, are reported in Fig. 3(b).

B. Experimental Setup

We compared our reinforcement learning NMPC (RL-NMPC) with two other state-of-the-art approaches, namely a standard NMPC [23] (with only the stage cost $\ell(\cdot)$ and the terminal cost matrix $\mathbf{P}$ but without the penalty function $B(\cdot)$), and the Artificial Potential Field (APF) [24]. All experiments were carried out using the robotic platform Locobot WX250s (Fig. 2(b)). To focus on the validation of the methodology alone and avoid confounding factors due to a field implementation, we executed high-fidelity simulations using ROS and Gazebo. Since the

thalamus, amygdala, and brainstem (NMPC) were previously implemented in MATLAB for training the SAC agent, we used MATLAB's ROS package to port these modules within the Locobot's ROS network. A crucial aspect for the optimal operation of the NMPC is the execution time required to solve the optimization problem. To avoid the overhead of running Gazebo and MATLAB simultaneously on a single laptop, we split the tasks between two different laptops. The first one, an Acer Nitro AN515-44 with an AMD Ryzen 7 CPU (2.90 GHz) and 8 GB of RAM, was dedicated to Gazebo simulations, while the second laptop, a Macbook Air M2 with 8 cores (4 at 3.20 GHz and 4 at 2.00 GHz) and 8 GB of RAM, was used to run the other components of the architecture, in particular the NMPCs that require specific solvers to solve the optimization problems. In contrast, the APF, which has low computational complexity and does not rely on optimization algorithms, was implemented in C++ and run on the same laptop as the simulations.

IV. Result and Discussion

For all the three algorithms, the sampling time is set to $T_s = 0.2$ s with control input constraints defined by the robot's speed limits, specifically $\mathbf{u}_{\min} = [v_{\min}, \omega_{\min}]^T = [0 \text{ m/s}, -1.5 \text{ rad/s}]^T$ and $\mathbf{u}_{\max} = [v_{\max}, \omega_{\max}]^T = [0.5 \text{ m/s}, 1.5 \text{ rad/s}]^T$. The RL-NMPC uses a prediction horizon of $N = 20$ (4 s), while the maximum weights for denormalizing the SAC agent's actions in the amygdala are set to $\mathbf{Q}_{\max} = \text{diag}(20, 20)$, $\mathbf{R}_{\max} = \text{diag}(10, 10)$, $\boldsymbol{\alpha}_{\max} = 100$, $\boldsymbol{\beta}_{\max} = 0.1$. The standard NMPC uses a prediction horizon of $N = 50$ steps (10 s) and weights set to $\mathbf{Q} = \text{diag}(15, 15)$, $\mathbf{R} = \text{diag}(5, 5)$ and $\mathbf{P} = \text{diag}(60, 60)$. For the APF, the robot is guided towards the goal by an attractive force $F_k^{att} = (v_{\max} \frac{d_k^{ref}}{\|d_k^{ref}\|} - v_k)/\tau$ and repelled by a repulsive force $F_k^{rep} = Ae^{(1-\frac{d_k^{obs}}{B})} \frac{d_k^{obs}}{\|d_k^{obs}\|}$ for each obstacle. Here, d^{ref} and d^{obs} are the goal and the considered obstacle distances, respectively, while $\tau = 0.5$, $A = 1$ and $B = 1$ are parameters that allow to tune the strength of the two forces. We evaluated the three approaches in two environments: one static and one dynamic, both characterized by obstacles with different danger levels. In both scenarios, the robot starts at $\mathbf{x}_0 = [0, 0]^T$ and aims to reach the goal position $\mathbf{x}_r = [6, 6]^T$, trying not only to avoid obstacles but also to ensure self-preservation. We recorded the path generated by each algorithm as well as their relative distances from the two obstacles. For the NMPCs, we also compared the execution time. To defined the NMPCs' constraints, obstacles are modeled as circles with a radius of $r_{obs} = 0.25$ m, which are consistent with the size of the robotic platform used in the experiment ($r_{robot} = 0.25$ m). Consequently, both NMPCs consider a minimum distance constraint of 0.5 m from the robot and each obstacle.

1) Static Environment: in this scenario, the dangerous and non-dangerous obstacles are placed at position $[2.5, 2.3]^T$ and at position $[5.2, 4.0]^T$, respectively. The results of the experiment are illustrated in Fig. 4(a). The APF's highly reactive nature leads to an irregular and inefficient path, characterized by approaches and retreats from obstacles caused by zones where the overall potential is nearly zero. Although the APF is able to maintain a minimum distance of 1.14 m from dangerous obstacles and 1.12 m from non-dangerous ones, it takes approximately 70 s to reach the goal position (Fig. 4(b)), significantly longer than two other approaches. Conversely, the conventional NMPC formulates a more straightforward trajectory toward the target, achieving a minimum clearance of 0.49 m from the hazardous obstacle (Fig. 4(b)), albeit with minor constraint infringements. Distinct from the other two methodologies, our RL-NMPC is systematically trained to attain the destination while ensuring an appropriate separation from obstacles, thereby diminishing the perceived fear level. This allows the robot to follow a smoother and safer path than the other methods, maintaining a minimum distance of 1.59 m from the dangerous obstacle and 2.18 m from the non-dangerous one.

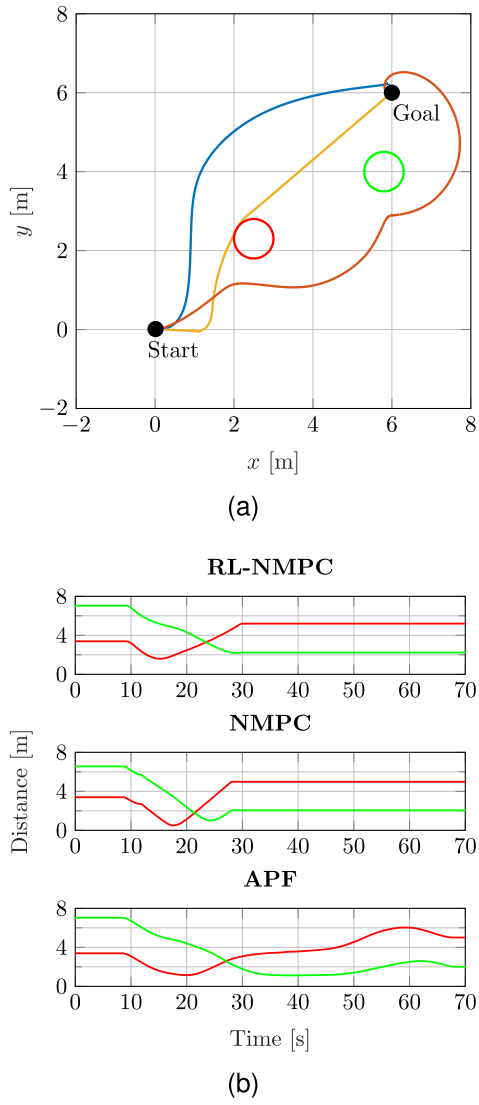

Fig. 4. Navigation results in static scenario: (a) path of the three implemented algorithms (RL-NMPC, NMPC, APF). - (b) distances between the robot and the two obstacles (Dang obs, Non-dang obs) for each of the considered algorithm.

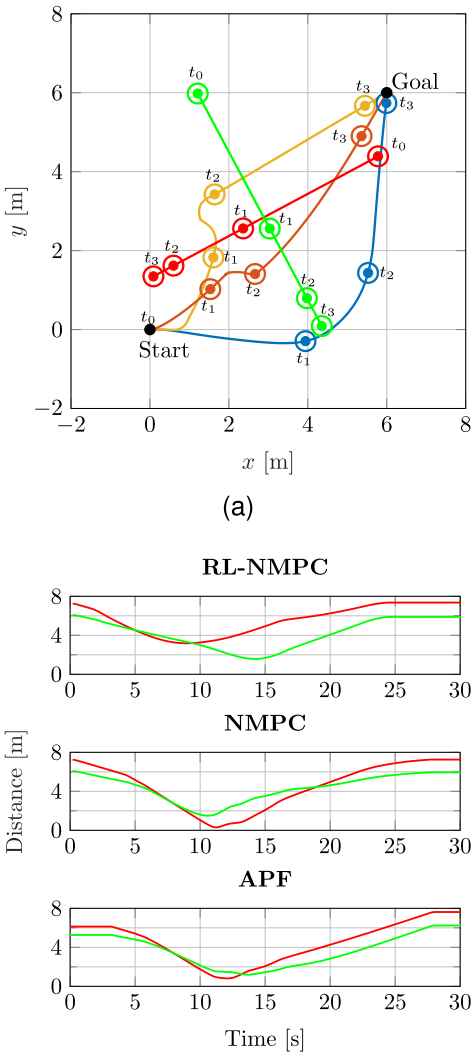

Fig. 5. Navigation results in dynamic scenario: (a) path of the three implemented algorithms (RL-NMPC, NMPC, APF) with corresponding positions at time instants $t_0 = 0$ s, $t_1 = 10$ s, $t_2 = 15$ s and $t_3 = 25$ s. - (b) distances between the robot and the two obstacles (Dang obs, Non-dang obs) for each of the considered algorithm.

2) Dynamic Environment: the dangerous obstacle starts at position $[5.78, 4.39]^T$ and moves linearly towards the goal located at $[0.08, 1.34]^T$, while the non-dangerous one starts at $[1.21, 5.97]^T$ and moves towards $[4.35, 0.08]^T$. Both obstacles are driven by a control law proportional to their distance from the targets, with a maximum speed of 0.4 m/s to match the robot's velocity and ensure meaningful interaction. At $k = 10$ s (Fig. 5(a)), the NMPC attempts to anticipate the movements of the dangerous obstacles but fails, reaching a minimum distance of 0.30 m (Fig. 5(b)) and resulting in a significant violation of the obstacle distance constraints. Although effective in static scenarios, this approach can lead to suboptimal paths in dynamic environments, posing safety risks to robots in real-world applications. The APF's risk of the robot getting stuck in regions with a very low resultant force is reduced by the moving obstacles that continuously alter the total potential field, resulting in a more directed path with a minimum distance of 0.80 m from dangerous obstacles and 1.18 m from non-dangerous one. In contrast, our RL-NMPC effectively differentiates obstacle dangerousness maintaining a minimum distance of 3.18 m from the dangerous obstacle and 1.57 m from non-dangerous one, while still reaching its goal, demonstrating superior self-preservation and adaptability also in dynamic environments.

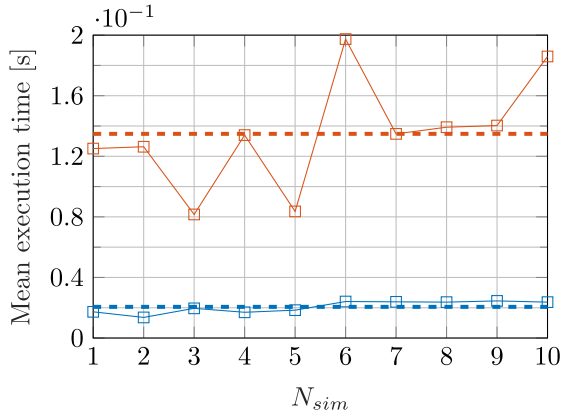

Fig. 6. Mean execution times of RL-NMPC and NMPC in ten random simulations with corresponding average mean execution times (0.0206 s and 0.1348 s, respectively).

3) Mean Execution Time: we evaluated the execution times of both NMPC implementations by running ten simulations in dynamic environments with randomized initial and final obstacle positions. For a fair comparison, we tested both NMPCs under identical conditions. NMPCs' performance was assessed by computing the mean execution time for each simulation, mitigating the impact of possible outliers in execution times. We also computed the average execution time over all simulations to quantify robustness across environments. Results shown in Fig. 6 indicate that while both implementations adhered to the sampling time constraint of $T_s = 0.2$ s, the RL-NMPC significantly outperformed the standard NMPC, with average execution times of 0.0206 s compared to 0.1348 s. This improvement is mainly due to the reinforcement learning framework, which guides convergence through the training process rather than relying on terminal constraints, as in traditional NMPC. This allows for a shorter prediction horizon N, which reduces the computational burden for the optimization problem. Furthermore, the dynamic adjustment of NMPC weights facilitated by the learning process permits the robot to maintain an increased distance from obstacles, thereby enhancing safety and simplifying the satisfaction of optimization constraints. Consequently, this methodology attains more rapid and stable solutions across varying scenarios, in contrast to the traditional approach, which demonstrates substantial variability in mean execution times.

V. Conclusions and Future Works

This letter presents a novel neuro-inspired control architecture for robotic navigation in hazardous environments. By capitalizing on the intrinsic mechanisms of the "Low Road" within the dual-pathway model, we improve adaptability and self-preservation during the execution of robotic navigation

tasks. Our approach improves traditional NMPCs and APFs by providing more generalizable, reliable, and smoother navigational capabilities. The adaptive training of NMPC parameters optimizes behavior while adhering to constraints, thus reducing random actions, simplifying the training process, and enhancing convergence. This results in fewer training episodes and better generalization compared to conventional RL methods, while providing superior efficiency and computational stability over standard NMPC implementations.

The investigation detailed in this letter represents an initial phase within a broader research trajectory, aiming not only to assess whether neuro-inspired control architectures exhibit advantages over conventional control frameworks, but also to establish connections between cognitive and control concepts by incrementally incorporating a diverse range of indicators for the self-preservation of robots – an artificial equivalent of primary emotions elicited in humans. This integration seeks to facilitate more robust, adaptive, and intelligent behaviors in robotic systems.

Although the proposed methodology offers several advantages, it is not without limitations. Specifically, it enforces minimum bounds on obstacle distances, which can impede navigation within confined spaces. Consequently, a trade-off may be required between ensuring self-preservation and satisfying task objectives. We intend to address this constraint through the development of a learning strategy centered on risk assessment [25]. Additionally, the randomization strategy in Eq. (10) does not account for variations in parameters related to the dangerousness of obstacles, limiting the architecture's ability to adapt to untrained dynamics of the fear level, although goal achievement and obstacle avoidance remain ensured. To tackle these challenges, future efforts will enhance the randomization strategy by including a parametrized version of the fear level and will implement the "High Road" pathway, as delineated in the LeDoux paradigm. To the latter aim, we envisage to utilize multimodal large language models (LLMs). In summary, the proposed architecture presents a promising solution for advanced robotics applications, including manipulation and navigation within human-shared environments. Specifically, the "Low Road" mechanism facilitates rapid adaptation to stimuli pertinent to the task, while the "High Road" will be dedicated to processing intricate information such as safety protocols, social norms, and strategic task planning. This dual-process approach is poised to significantly advance the development of intelligent systems capable of operating efficiently and safely in dynamic and unpredictable scenarios.

Acknowledgment

This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] Y. Ham, K. K. Han, J. J. Lin, and M. Golparvar-Fard, "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVs): A review of related works," *Visual. Eng.*, vol. 4, no. 1, pp. 1–8, Jan. 2016.

[2] T. Tomic et al., "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[3] G. Galati, S. Primatesta, S. Grammatico, S. Macrì, and A. Rizzo, "Game theoretical trajectory planning enhances social acceptability of robots by humans," *Sci. Reports*, vol. 12, no. 1, Dec. 2022, Art. no. 21976.

[4] K. Man and A. Damasio, "Homeostasis and soft robotics in the design of feeling machines," *Nature Mach. Intell.*, vol. 1, no. 10, pp. 446–452, Oct. 2019.

[5] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 3. Pusan, South Korea, 2001, vol. 3, pp. 1530–1535.

[6] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Proc. 10th Int. Symp. Distrib. Auton. Robot. Syst.*, Berlin, Germany, 2013, pp. 203–216.

[7] F. Micheli, M. Bersani, S. Arrigoni, F. Braghin, and F. Cheli, "NMPC trajectory planner for urban autonomous driving," *Vehicle Syst. Dyn.*, vol. 61, no. 5, pp. 1387–1409, 2023.

[8] B. Lindqvist, S. S. Mansouri, A.-A. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6001–6008, Oct. 2020.

[9] L. Calogero, M. Mammarella, and F. Dabbene, "Learning model predictive control for quadrotors minimum-time flight in autonomous racing scenarios," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1063–1068, 2023.

[10] D. Kostadinov and D. Scaramuzza, "Online weight-adaptive nonlinear model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, Las Vegas, NV, USA, 2020, pp. 1180–1185.

[11] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments," *J. Intell. Robotic Syst.*, vol. 98, pp. 297–309, Sep. 2020.

[12] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 3517–3524.

[13] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020.

[14] R. Plutchik, "A Psychoevolutionary Theory of Emotions," *Social Sci. Inf.*, vol. 21, no. 4, pp. 3–33, Jul. 1982.

[15] L. Pessoa, "Do intelligent robots need emotion," *Trends Cogn. Sci.*, vol. 21, no. 11, pp. 817–819, Nov. 2017.

[16] L. Cañamero, "Emotion understanding from the perspective of autonomous robots research," *Neural Networks*, vol. 18, no. 4, pp. 445–455, May 2005.

[17] T. Watt Smith et al., *The Book of Human Emotions: An Encyclopedia of Feeling From Anger to Wanderlust*. London, U.K.: Wellcome Collection, 2015.

[18] J. E. LeDoux, "Sensory systems and emotion: A model of affective processing," *Integr. Psychiatry*, vol. 4, no. 4, pp. 237–243, 1986.

[19] P. Den Dulk, B. T. Heerebout, and R. H. Phaf, "A computational study into the evolution of dual-route dynamics for affective processing," *J. Cogn. Neurosci.*, vol. 15, no. 2, pp. 194–208, Feb. 2003.

[20] E. T. Rolls, "Neurophysiology and functions of the primate amygdala," In J. P. Aggleton (Ed.), The amygdala: Neurobiological aspects of emotion, memory, and mental dysfunction, Wiley-Liss, 1992, pp. 143–165.

[21] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 1861–1870.

[22] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, precision, and payloads: Adaptive nonlinear MPC for quadrotors," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 690–697, Apr. 2022.

[23] M. Sani, B. Robu, and A. Hably, "Dynamic obstacles avoidance using nonlinear model predictive control," in *Proc. IEEE Annu. Conf. Ind. Electron. Soc.*, Toronto, ON, Canada, 2021, pp. 1–6.

[24] A. Usai, "Modelling and simulation of mobile robot motion and its interaction with humans," M.Eng. thesis, Politecnico di Torino, Turin, Italy, 2023. [Online]. Available: https://webthesis.biblio.polito.it/28557/

[25] S. Primatesta, G. Guglieri, and A. Rizzo, "A risk-aware path planning strategy for UAVs in urban environments," *J. Intell. Robot. Syst.*, vol. 95, pp. 629–643, Sep. 2019.