Scuola IMT Alti Studi Lucca

Human-like trajectories generation via receding horizon tracking applied to the TickTacking interface

Questa è la versione sottoposta a revisione paritaria (postprint) della seguente opera:

Original
Human-like trajectories generation via receding horizon tracking applied to the TickTacking interface / Masti, D., Menchetti, S., Erdem, C., Gnecco, G.S., Rocchesso, D.. - (2025), pp. 1476-1481. (IEEE SMC 2025 - 2025 IEEE International Conference on Systems, Man, and Cybernetics Vienna, Austria 5-8/10/2025) [10.1109/SMC58881.2025.11342906].

Availability:
This version is available at: 20.500.11771/38102

Publisher:

Published
DOI:10.1109/SMC58881.2025.11342906

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

09 November 2026

arXiv:2507.13528v1 [cs.HC] 17 Jul 2025

Human-Like Trajectories Generation via Receding Horizon Tracking Applied to the TickTacking Interface

Daniele Masti[1], Stefano Menchetti[2], Çağrı Erdem[3], Giorgio Gnecco[2], and Davide Rocchesso[3‡]

July 21, 2025

Abstract

TickTacking is a rhythm-based interface that allows users to control a pointer in a two-dimensional space through dual-button tapping. This paper investigates the generation of human-like trajectories using a receding horizon approach applied to the TickTacking interface in a target-tracking task. By analyzing user-generated trajectories, we identify key human behavioral features and incorporate them in a controller that mimics these behaviors. The performance of this human-inspired controller is evaluated against a baseline optimal-control-based agent, demonstrating the importance of specific control features for achieving human-like interaction. These findings contribute to the broader goal of developing rhythm-based human-machine interfaces by offering design insights that enhance user performance, improve intuitiveness, and reduce interaction frustration.

This is the authors' version of a paper accepted at *2025 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. It is posted here for your personal use, not for redistribution.
Please cite the conference version once available.
D. Masti, S. Menchetti, Ç. Erdem, G. Gnecco and D. Rocchesso, "Human-Like Trajectories Generation via Receding Horizon Tracking Applied to the TickTacking Interface" in *2025 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, 2025.*

*[1]Daniele Masti is with Gran Sasso Science Institute, Viale F. Crispi 7, 67100, L'Aquila, Italy daniele.masti@gssi.it

†[2]Stefano Menchetti and Giorgio Stefano Gnecco are with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy {stefano.menchetti, giorgio.gnecco}@imtlucca.it

‡[3]Çağrı Erdem and Davide Rocchesso are with the University of Milan, Via Celoria 18, 20133, Milan, Italy {cagri.erdem, davide.rocchesso}@unimi.it

You may use the following bibtex entry:

```
@inproceedings{masti2025human,
title = {Human-Like Trajectories Generation via Receding Horizon Tracking Applied to the TickTacking Interface},
author= {Masti, Daniele and Menchetti, Stefano and Erdem, Çağrı and Gnecco, Giorgio and Rocchesso, Davide},
booktitle={2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)},
year = {2025},
organization = {IEEE}
}
```

1 INTRODUCTION

Human-Computer Interfaces (HCIs) rely on multiple input and output modalities to facilitate interaction between users and computers [1]. In most cases, they convey information through visual and auditory signals, while receiving input via keyboards, buttons, and other pressure-based methods. However, interaction is not limited to these conventional inputs, as variations in timing, rhythm, and input sequences can also serve as valuable sources of information.

A notable example of rhythm-based HCI is the *TickTacking* interface, introduced in [2] and further detailed in [3]. This interface falls within the broader research area of rhythmic interaction in human-computer interaction, where rhythm serves as a communication channel between users and computers [4]. In [3], the TickTacking interface was tested in a gamified scenario where users tracked an object on the screen using a pointer controlled by rhythmic tapping on two buttons. Although this study highlighted high user engagement, the effectiveness of the interface in conveying precise commands has not been systematically evaluated. This work addresses that gap by comparing an optimal-control-based agent designed for efficiency with an artificial agent that mimics human behavior when using the TickTacking interface.

Focusing on designing a controller that exhibits human-like behavior rather than directly analyzing human trajectories serves several key purposes. First, it provides insight into the trade-offs, constraints, and objectives that shape human interaction with the TickTacking interface. These insights can inform interface improvements and the development of user guidance systems that improve performance and intuitiveness, and reduce frustration by providing visual or auditory feedback. In addition, such guidance systems can be integrated into a cyclic learning process, where they help users refine their control strategies, generating new data that are leveraged to improve the guidance systems. Further, a human-inspired controller enables the prediction of user outcomes in

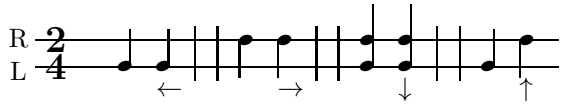


Figure 1: Encoding of controlled velocity directions, in case duplets are used by the TickTacking interface.

novel but related tasks on the same interface. Ultimately, this research contributes to the broader goal of identifying fundamental principles for designing general-purpose rhythmic HCIs, building on insights from previous works such as [5–8].

This paper is organized as follows: § 2 provides an overview of the TickTacking interface, explaining its functionality and how human-generated trajectories are collected. We analyze these trajectories in two scenarios: one in which users freely navigate and the other in which they perform a target-tracking task, extracting key behavioral features from both. § 3 describes the receding horizon control approach used to generate artificial trajectories that preserve essential human characteristics while solving the tracking problem. Finally, results are presented in § 4, followed by conclusions and future research directions in § 5.

2 TickTacking Interface: Background and Human Features Extraction

TickTacking investigates rhythmic human control behavior through an interface consisting of a visual display and a dual-button input device, with each button assigned to one hand. Users control the direction and magnitude of a pointer's velocity on the interface screen by varying the combination and rate of their taps. In [3], the use of *duplets* and *triplets* – i.e., rhythmic patterns consisting of two or three taps per button within the same rhythmic cell – was examined. When no tapping occurs, the pointer maintains a constant velocity. The interface design constrains movement by allowing only one velocity coordinate to change at a time. As an example, Fig. 1 illustrates, using music notation, how different combinations of Right (R) and Left (L) button taps determine the controlled component (and its sign) of the pointer's velocity vector when encoded as duplets. In this encoding, the Inter-onset Interval (IoI) – the time between two successive taps within the same duplet – modulates the magnitude of the controlled velocity component at the end of each duplet. Specifically, the absolute value of the controlled velocity component is inversely proportional to the IoI.

In [3], the effectiveness of the TickTacking interface was evaluated in two experiments with ten and seventeen participants, respectively. These experiments had different objectives: the first assessed controllability and user engagement, while the second examined interface usage under partial visual feedback deprivation. Given the additional complexity of visual feedback deprivation and

the challenge of modeling an artificial controller to account for it, the present study focuses on the results of the first experiment from [3]. In the first experiment, ten participants first completed a training phase where they freely explored the TickTacking interface to navigate a two-dimensional space. They then performed a target-tracking task. Each participant used both versions of the interface: one utilizing duplets and the other triplets. Due to incomplete recordings of free navigation sessions, the final dataset consists of 16 free navigation trajectories and 20 tracking trajectories. Analysis of this data revealed the following key insights:

2.0.1 Velocity Direction

As pointed out in [3], the histogram of velocity directions reveals that users predominantly chose specific directions, likely due to the ease of generating certain movements. Specifically, Fig. 2 shows that users favored directions at 45°, 135°, 225° or 315° relative to the positive horizontal axis in both free navigation and target tracking tasks. This suggests a preference for equal-length tapping time intervals along the horizontal and vertical axes. This preference aligns with findings in related fields such as animal locomotion [9], vocal communication [10], and music cognition [11], where simple temporal ratios (e.g., $1:1$, $1:2$, $2:1$) are commonly favored. These results indicate that human motor behavior in rhythmic control tasks is influenced by fundamental timing principles observed across different domains.

2.0.2 Velocity Magnitude

Another key human-related feature, already considered in [3], is velocity magnitude, measured in pixels per second and originally defined as the Euclidean norm of the velocity vector. However, in this work, we instead define the velocity magnitude using its l_1 norm, which better aligns with the TickTacking interface, as velocity changes occur coordinate-wise. Additionally, the l_1 norm has the favorable numerical characteristic of being linear in the velocity components. Fig. 3(a) presents a Kernel Density Estimate (KDE) of the velocity l_1 norm for both the free navigation and target tracking scenarios. For analysis purposes, we approximate the free navigation KDE with its best-fit Gaussian Probability Density Function (PDF), itself illustrated in Fig 3(a) with a dash-dot line. This Gaussian PDF closely aligns with the free navigation KDE while providing a more convenient formulation. We observe that the KDE of the human tracking data is a trade-off between the reference KDE and the Gaussian PDF, balancing the goal of perfect tracking with inherent human limitations. This suggests that humans strive to adapt their natural tapping tendencies [12] to achieve acceptable tracking performance.

2.0.3 Control Sparsity

In [3], human-generated trajectories using the TickTacking interface exhibited a characteristic zig-zag pattern. This pattern arises from the interface's design,

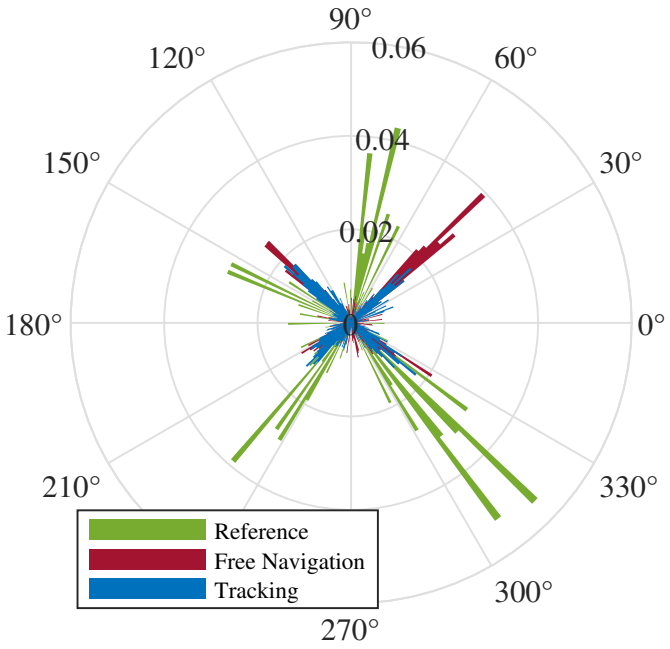
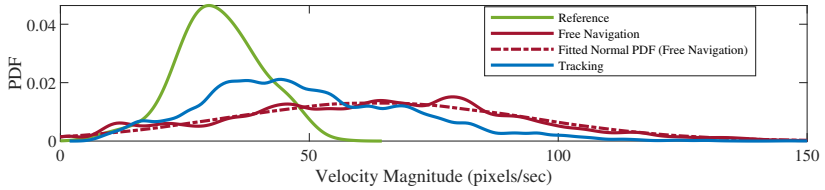


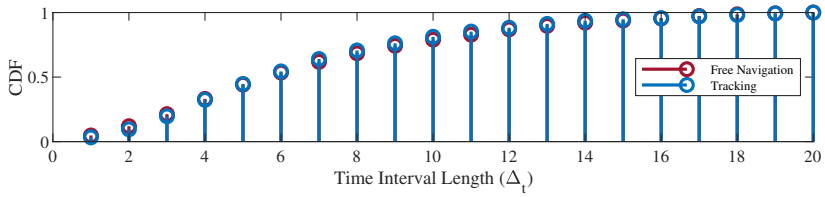
Figure 2: Histogram of velocity directions for the free navigation and target tracking scenarios in [3], compared to the velocity directions of the reference trajectory. The bar height represents the relative frequency of observations.

its technical characteristics, and the dexterity required for effective operation, leading to both spatial and temporal sparsity in user input. We have already discussed the cause of spatial sparsity: the interface allows the user to affect only one velocity component at a time. Temporal sparsity, on the other hand, stems from the infrequent nature of user actions due to both biomechanical constraints and the time required by the users to observe the outcome of their previous action before choosing the next one. Fig. 3(b) illustrates this feature through the empirical Cumulative Density Function (CDF) $\hat{F}_T$ of the time interval between input actions in the free navigation case. The time interval is measured in sample intervals $\Delta_t \in \mathbb{R}$, where $\Delta_t = 0.283$ sec. The CDF is defined over the support $\{\Delta_t, \ldots, 20\Delta_t\}$, with low-probability larger time intervals removed for refinement. Additionally, Fig. 3(b) presents the corresponding empirical CDF for the target tracking scenario, which closely aligns with the free navigation CDF. This minimal divergence suggests that the interval length between control actions is a fundamental human property unaffected by the specific task or objective.

These findings define the ideal features for trajectories generated by a human-inspired artificial controller. As a result, these characteristics will inform the design of the optimization-based controller, guiding the formulation of its penalty terms and constraints.



(a) KDEs of the velocity l_1 norm in the free navigation and target tracking scenarios, compared to the KDE of the reference trajectory. The Gaussian PDF that best fits the free navigation data is also shown for comparison.



(b) Empirical CDF of the time intervals between control action updates in the free navigation and target tracking scenarios.

Figure 3: Features of the human-generated trajectories acquired in [3] (in addition to those shown in Fig. 2).

3 Human-Inspired Trajectory Generation Using Receding Horizon Tracking

The TickTacking tracking task can be seen as a control problem in which the user acts as a controller, seeking actions that best allow the pointer to follow the target. From this standpoint, a natural way to generate human-like artificial trajectories is to replace the user with a control algorithm designed to mimic human behavior as closely as possible. A suitable approach for this constrained control problem is the Receding Horizon (RH) controller, also known as a Model Predictive Control (MPC) [13] algorithm.

The RH technique is a recursive strategy. In a discrete-time framework, where N is the final decision stage, and $W \in \{1, \dots, N\}$ is the prediction window length (a key hyperparameter), the controller solves a constrained optimization problem at each time instant $t = n\Delta_t$, with $n \in \{1, \dots N - W + 1\}$, based on the current state $\mathbf{s}(t)$ of the dynamical system. This optimization determines the optimal control sequence $\mathbf{U}^\star(t) = \{\mathbf{u}^\star(t), \mathbf{u}^\star(t + \Delta_t), \dots, \mathbf{u}^\star(t + (W-1)\Delta_t)\}$. Then, the first control action $\mathbf{u}^\star(t)$ is applied, updating the system state to $\mathbf{s}(t + \Delta_t)$, the time window shifts forward by one unit (it "recedes"), and the process repeats.

RH controllers approximate the optimal solution of a *global* optimization problem that would involve an infinite prediction window. However, solving such a problem is generally infeasible due to its infinite number of optimization variables. Indeed, balancing the length of the prediction window is a delicate act that must optimize the trade-off between achieving long-term objectives

and guaranteeing computational tractability. Moreover, for the specific case addressed in this work, limiting the prediction horizon ensures that the algorithm does not gain an unrealistic advantage over human users, who typically predict future trajectory evolution within a finite time span.

Since RH controllers are *model-based*, designing the controller first requires the definition of a kinematic model, specifying the state and action variables and their relationship through the state transition function. We consider a simple discrete-time state-space kinematic model to describe the motion of the pointer on the screen. The state $\mathbf{s} \in \mathbb{R}^2$ consists of the horizontal and vertical coordinates x and y of the pointer's position, while the control action $\mathbf{u} \in \mathbb{R}^2$ represents the velocity vector. The state transition function is

$$\mathbf{s}(t + \Delta_t) = \mathbf{s}(t) + \mathbf{u}(t)\,\Delta_t\,. \tag{1}$$

For simplicity, we assume the sampling interval Δ_t corresponds to the time between any two successive possible control updates. The duration of Δ_t is assumed to be large enough to contain a single tap combination (duplets or triplets), if any, on the TickTacking interface, eliminating the need to explicitly distinguish between these rhythmic elements. Additionally, since the TickTacking interface allows for no control update (when no button is pressed), we introduce an additional action variable $\mathbf{u}_{\text{old}} \in \mathbb{R}^2$ to retain the previous value of the controlled velocity. This variable is updated according to $\mathbf{u}_{\text{old}}(t + \Delta_t) = \mathbf{u}(t)$.

Leveraging the variables that we have introduced, at each time step t, the RH controller solves the following mathematical programming problem:

$$\begin{aligned} \min_{\mathbf{U}(t),\mathbf{S}(t),\mathbf{d}_s(t)} \; & \text{MSE}(\mathbf{S}(t), \mathbf{S}_{\text{ref}}(t)) + \sum_{k=1}^{N_p} \lambda_k \mu_k(\mathbf{U}(t))\,, \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{S}(t), \mathbf{U}(t), \mathbf{U}_{\text{old}}(t), \mathbf{d}_s(t)\right) \leq \mathbf{0}\,, \\ & \mathbf{h}\left(\mathbf{S}(t), \mathbf{U}(t), \mathbf{U}_{\text{old}}(t), \mathbf{d}_s(t)\right) = \mathbf{0}\,. \end{aligned} \tag{2}$$

Here, $\mathbf{U}(t) = \{\mathbf{u}(t), \ldots, \mathbf{u}(t + (W-1)\Delta_t)\}$ represents the control sequence over the prediction window, $\mathbf{U}_{\text{old}}(t)$ is the corresponding sequence of $\mathbf{u}_{\text{old}}$, $\mathbf{S}(t) = \{\mathbf{s}(t), \ldots, \mathbf{s}(t + W\Delta_t)\}$ is the state sequence and $\mathbf{S}_{\text{ref}}(t)$ is the corresponding state sequence of the reference trajectory. The sequence $\mathbf{d}_s(t) = \{d_s(t), \ldots, d_s(t + (W-1)\Delta_t)\}$, with $d_s \in \mathbb{R}$, represents a sequence of additional decision variables to enforce specific controller features.

The objective function includes the Mean Squared Error (MSE) term

$$\text{MSE}\left(\mathbf{S}(t), \mathbf{S}_{\text{ref}}(t)\right) \triangleq \frac{1}{W} \sum_{k=1}^{W} \|\mathbf{s}(t + k\Delta_t) - \mathbf{s}_{\text{ref}}(t + k\Delta_t)\|_2^2\,, \tag{3}$$

along with N_p penalty terms μ_k, each weighted by a non negative scalar λ_k, to enforce desired controller behaviors. The notation $\|\cdot\|_2$ denotes the Euclidean or l_2 norm. The constraints, represented by the vector-valued functions $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$, ensure that the controller satisfies certain conditions. Fundamental

inequalities constraints bound the state components to the TickTacking interface dimensions, enforcing $0\,\text{pixels} \leq x \leq 1920\,\text{pixels}$ and $0\,\text{pixels} \leq y \leq 1080\,\text{pixels}$. Among the key equality constraints are the kinematic system (1) and the update rule for $\mathbf{u}_{\text{old}}$.

3.1 Human-Like RH Control

The optimal solution to the trivial RH target tracking problem (2), constrained only to the kinematic system and the interface dimensions, and without penalty terms, produces a tracking trajectory that, after the first time step, aligns with the target trajectory, but deviates significantly from human motion. Consequently, this solution serves as a baseline for developing the human-inspired RH controller by introducing specific human-related penalty terms, constraints, and modifications to the control strategy.

Following the order in § 2, the first human-related feature we consider is the distribution of the velocity directions. Since the most common directions in human motion are 45°, 135°, 225° and 315° – where velocity components have equal magnitude – we introduce the penalty term:

$$\mu_1(\mathbf{U}(t)) = \frac{1}{W} \sum_{k=0}^{W-1} \Big| |v_x(t+k\Delta_t)| - |v_y(t+k\Delta_t)| \Big|.$$

We recall that v_x and v_y are the two components of the control $\mathbf{u}$. Then, the term μ_1 penalizes deviations from the preferred diagonal directions, thereby encouraging human-like directional movement.

Another important characteristic of human motion is the magnitude of velocity, represented by its ℓ_1 norm. Based on the Gaussian described in § 2 and shown in Fig. 3(a) – with mean 63.75 pixels/sec and standard deviation 30.45 pixels/sec – we define the penalty term μ_2 as the following approximation of its negative log formulation:

$$\mu_2(\mathbf{U}(t)) = \frac{1}{W} \sum_{k=0}^{W-1} (\|\mathbf{u}(t+k\Delta_t)\|_1 - 63.75)^2.$$

This approximation is twofold: with respect to the negative log formulation of the Gaussian PDF, we neglect the additional constant term $-\log(\sqrt{2\pi}\sigma_{|\mathbf{u}|})$ and drop the squared standard deviation at the denominator of μ_2. By combining this term with μ_1, we ensure that velocity magnitudes remain within human-like ranges.

The last features to address concern the spatial and temporal sparsity of the human control, affecting the direction and frequency of the control updates, respectively. We begin by taking into account the spatial sparsity. To model spatial sparsity, we introduce the integer decision variable $d_s \in -1, 0, 1$, enforcing the update of only one velocity component at a time via the following conditional constraints:

$$\begin{cases} d_s \leq 0.9 & \Rightarrow v_x = v_{x,\text{old}}\,, \\ d_s \geq -0.9 & \Rightarrow v_y = v_{y,\text{old}}\,, \end{cases}$$

which bind the velocity update to the specific value of the corresponding d_s. These constraints ensure that velocity updates occur only along one axis at a time, also introducing a rudimentary temporal sparsity. To enforce temporal sparsity, we modify the control strategy to obtain a stochastic event-triggered formulation. Specifically, the optimization problem (2) is solved and the control $\mathbf{u}$ updated only if the stochastic condition

$$d_t \leq \hat{F}_T(t_{\text{event}}) \tag{4}$$

is met. Here, $\hat{F}_T$ is the CDF described in § 2 and shown in Fig. 3(b). The variable $d_t \in [0, 1]$ is a uniformly distributed indicator variable, sampled at the first time instant and at any time instant in which the control action is actually updated. The variable $t_{\text{event}} \in \{\Delta_t, 2\Delta_t, \ldots, 20\Delta_t\}$ represents the time intervals since the last control action update, with its domain corresponding to the support of $\hat{F}_T$. Condition (4) ensures that the RH control strategy has an overall CDF of t_{event} coincident with $\hat{F}_T$, thereby mimicking the temporal limitations associated with the human behavior.

Additionally, the length of the prediction window plays a crucial role in the controller's temporal characterization, ensuring it does not outperform human users unfairly. We set the prediction window length to $8\Delta_t$, the integer closest to the mean of $\hat{F}_T$. In this way, the prediction window is, on average, equivalent to the time interval in which the control actions remain unchanged. This reasoning supports the introduction of the constraints

$$\mathbf{u}(t) = \mathbf{u}(t + \Delta_t) = \ldots = \mathbf{u}(t + (W - 1)\Delta_t)$$

which guarantee the identity between the elements of $\mathbf{U}(t)$.

Finally, we observe that the formulation of the human-like RH controller, which incorporates all the key human features discussed in § 2 through additional penalty terms and constraints, still maintains a simple theoretical interpretability. Moreover, from a computational perspective, the overall optimization problem remains quadratic. The use of the ℓ_1 norm for velocity magnitude and the structured constraint design ensures that the problem remains efficiently solvable. This allows us to leverage specialized solvers such as `GUROBI` for the rapid optimization of (2).

4 Numerical Results

We analyze two scenarios: a baseline scenario, where we present the baseline RH controller's results, and a human-like tracking scenario, where we examine the performance of the human-inspired RH controller described in § 3.1. To match the dataset size in [3], we generate 20 artificial trajectories in both scenarios initialized as each human-generated one.

The baseline scenario illustrates the behavior of the basic RH controller, which minimizes the MSE (3) at each time step, subject only to the kinematic system dynamics and display boundaries. In contrast, the second scenario explores the case where we enforce the human-inspired features described in § 3

λ_1	λ_2	σ_ε
32	0.6	2 pixels/sec

Table 1: Tuned parameters of the Human-Like RH Controller

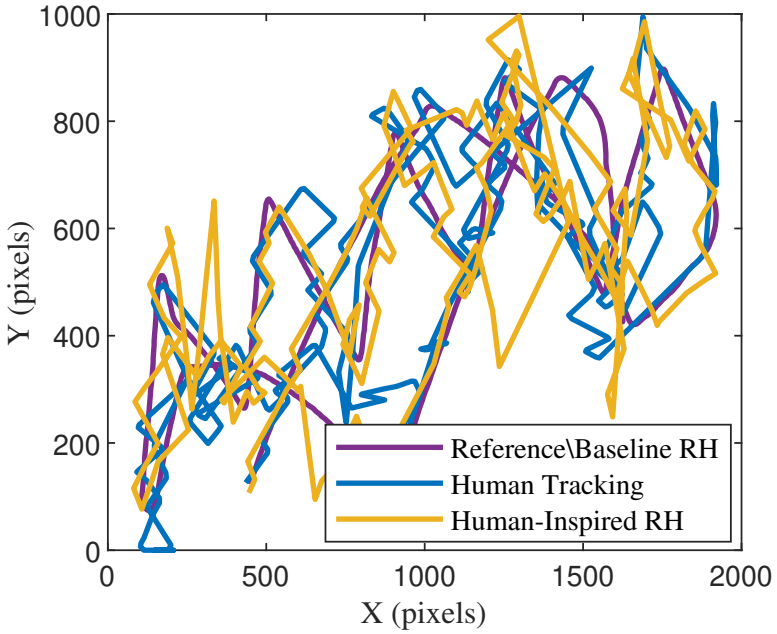


Figure 4: Example trajectories generated by the baseline and human-inspired RH controllers. The controller and reference trajectories practically overlap in the baseline scenario. One example of human-generated trajectories is included for comparison. A dynamic visualization of these trajectories is available at https://doi.org/10.5281/zenodo.15240338

to define the human-like RH controller. Table 1 presents the manually tuned parameters associated with such controller: λ_1 and λ_2 are the weights of the penalty terms, and σ_ε is the standard deviation of a Gaussian distributed, zero-mean noise added to the controlled action to account for residual stochasticity in human behavior not explicitly modeled by our formulation. The tuning process follows a trial-and-error approach aimed at replicating the observed statistical features of human behavior.

The results of the baseline scenario are shown in Fig. 4, Fig. 5 and Fig. 6. The velocity directions and l_1 norms of the baseline RH controller align closely with those of the reference trajectory. All generated trajectories quickly converge to the reference trajectory, achieving near-complete overlap as early as the second time step. The KDE of the MSE is impulsive, with the peak at zero.

The same figures also depict the results of the human-like scenario, showing

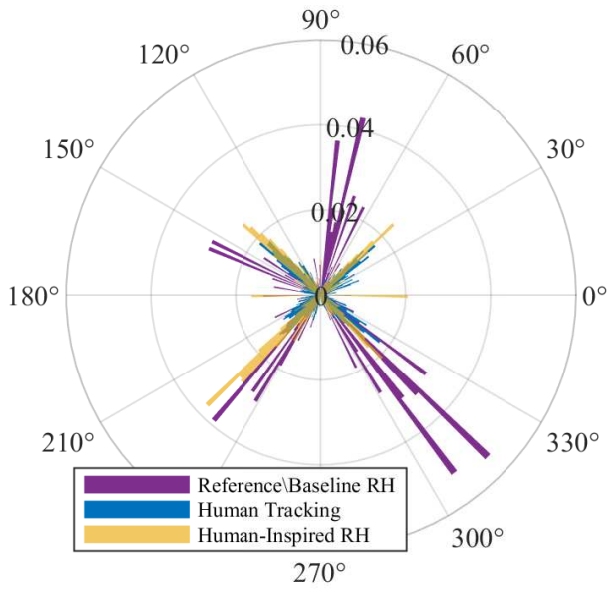
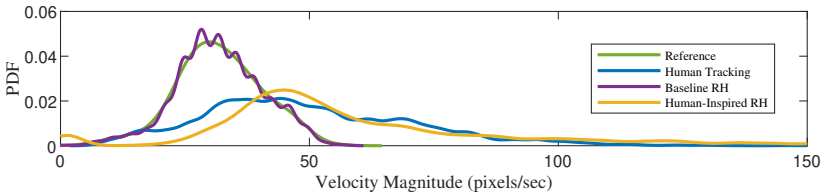


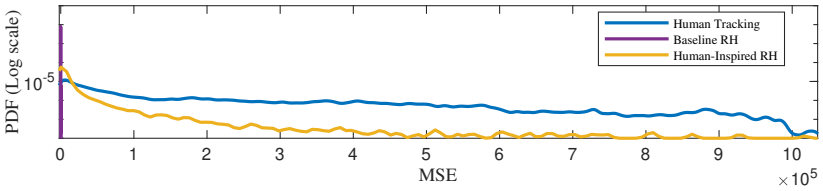
Figure 5: Histogram of velocity directions in the baseline RH and human-like RH scenarios, compared to the velocity directions in the human tracking scenario and the reference trajectory. The velocity directions of the reference trajectory and the baseline RH scenario overlap. Bars corresponding to the RH cases are partially transparent to enhance the visibility of the other two cases. The bar height represents the relative number of observations.

the effect of the penalty terms μ_1 and μ_2. They facilitate the generation of velocities that closely resemble those observed in human-generated trajectories for the same tracking task. When combined with the sparse control strategy, they enable the controller to replicate the human behavior, deviating from the exact tracking exhibited by the baseline RH controller. This affects coherently the KDE of the MSE, which still has a peak at zero but lacks the impulsive behavior seen in the baseline case. Nevertheless, the KDE of the MSE indicates that the human-like RH controller generally achieves lower MSE values compared to the empirical PDF of human-generated trajectories.

These findings suggest that the human-like RH controller is a viable approach for generating trajectories that resemble human behavior while demonstrating improved performance. As a result, this controller may serve as a valuable tool for designing new experiments to further investigate human behavior within the TickTacking interface, and to assist human users in decision-making and training applications.



(a) KDEs of the velocity l_1 norm in the baseline RH and human-like RH scenarios. The KDEs of the human tracking scenario and the reference trajectory are shown for comparison.



(b) KDEs of the MSE in the baseline RH, human-like RH, and human tracking scenarios.

Figure 6: Features of the RH-generates trajectories (in addition to those shown in Fig. 5).

5 Conclusions and Further Extensions

This study investigated the generation of human-like trajectories using an RH approach applied to the TickTacking interface, a novel rhythm-based HCI method. By analyzing user-generated trajectories, we identified key human-related features such as velocity directions, velocity magnitude PDF, and the probabilistic characterization of control sparsity. These features were used to design a controller that mimics human behavior, providing insights into the trade-offs, constraints, and objectives that humans navigate when interacting with this interface. While relying on the modeling of human-likeness introduced in this work for the specific TickTacking task, the control design procedure is generalizable to other tasks where specific human-like features can be defined and exploited. Our findings contribute to the broader goal of developing general-purpose rhythm-based human-machine interfaces. By understanding and replicating human behavior in control systems, it is possible to design more intuitive and effective user guidance systems that enhance overall interaction quality.

Future research could explore additional human-related features and extend the current analysis, potentially involving methods like Granger's causality [14] to assess the effectiveness in performing the tracking task. Furthermore, data-driven approaches could reduce computational demands by allowing to learn the prediction over the RH control horizon [15], or by tuning a lightweight model reference Linear Quadratic Regulator (LQR) controller to approximate RH performance [16]. Additionally, techniques from mixed-integer optimization [17] could be adapted to enhance the efficiency of the optimization step within the

control problem. Finally, a closed user-controller loop could be envisioned, wherein the controller adapts to the user while the user concurrently learns from the controller.

References

[1] J. Gupta, M. Ahuja, and N. Garg, "A review on human-computer interaction (hci)," in *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, 2022, pp. 1–6.

[2] D. Rocchesso, A. Bellino, and A. Perez, "TickTacking–drawing trajectories with two buttons and rhythm," in *20th Sound and Music Computing Conf.*, 2023, pp. 63–71.

[3] D. Rocchesso, A. Bellino, G. Ferrara, and A. Perez, "Spacetime trajectories as overlapping rhythms," *Int. Journal of Human-Computer Studies*, vol. 192, p. 103358, 2024.

[4] C. Erkut, A. Jylhä, and D. Rocchesso, *Heigh ho: Rhythmicity in sonic interaction.* MIT Press, 2013, ch. 17, pp. 341–350.

[5] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *Transactions on Control Systems Technology*, 2013.

[6] A. M. R. Lazcano, T. Niu, X. C. Akutain, D. Cole, and B. Shyrokau, "MPC-based haptic shared steering system: A driver modeling approach for symbiotic driving," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1201–1211, 2021.

[7] L. Cecchin, A. Trachte, L. Fagiano, and M. Diehl, "Real-time prediction of human-generated reference signals for advanced digging control," in *20th Int. Conf. on Automation Science and Engineering.* IEEE, 2024, pp. 496–501.

[8] S. Luo *et al.*, "Human–robot shared control based on locally weighted intent prediction for a teleoperated hydraulic manipulator system," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4462–4474, 2022.

[9] L. Laffi, T. Raimondi, C. Ferrante, E. Pagliara, A. Bertuglia, E. F. Briefer, M. Gamba, and A. Ravignani, "The rhythm of horse gaits," *Annals of the New York Academy of Sciences*, vol. 1543, no. 1, pp. 86–93, 2025.

[10] C. De Gregorio, D. Valente, T. Raimondi, V. Torti, L. Miaretsoa, O. Friard, C. Giacoma, A. Ravignani, and M. Gamba, "Categorical rhythms in a singing primate," *Current Biology*, vol. 31, no. 20, pp. R1379–R1380, 2021.

[11] N. Jacoby, R. Polak, J. A. Grahn, D. J. Cameron, K. M. Lee, R. Godoy, E. A. Undurraga, T. Huanca, T. Thalwitzer, N. Doumbia *et al.*, "Commonality and variation in mental representations of music revealed by a cross-cultural comparison of rhythm priors in 15 countries," *Nature Human Behaviour*, vol. 8, no. 5, pp. 846–877, 2024.

[12] B. H. Engler, A. Zamm, and C. Møller, "Spontaneous rates exhibit high intra-individual stability across movements involving different biomechanical systems and cognitive demands," *Scientific Reports*, vol. 14, no. 1, p. 14876, 2024.

[13] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: Theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[14] G. Gnecco, A. Camurri, C. Gasparotti, E. Ceccaldi, G. Volpe, B. Bardy, M. Bieńkiewicz, and S. Janaqi, "Measuring cues of leadership, cohesion, and fluidity in joint full-body movement to support embodied interaction design: A pilot study," *Human Behavior and Emerging Technologies*, vol. 2024, no. 1, p. 1636854, 2024.

[15] D. Masti, F. Smarra, A. D'Innocenzo, and A. Bemporad, "Learning affine predictors for mpc of nonlinear systems via artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5233–5238, 2020.

[16] D. Masti, M. Zanon, and A. Bemporad, "Tuning lqr controllers: A sensitivity-based approach," *IEEE Control Systems Letters*, vol. 6, pp. 932–937, 2021.

[17] M. Zamponi, E. Incerto, D. Masti, and M. Tribastone, "Certified inductive synthesis for online mixed-integer optimization," in *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2025)*, 2025, pp. 1–11.