Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-6 (2022) 502–507

Collision-free trajectory design for dance choreography of virtual drones in hierarchical structure ⋆

Balázs Németh *, Attila Lelkó *, Zoltán Antal *, Ajtony Csaba **

* *Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH), Kende u. 13-17, H-1111 Budapest, Hungary. E-mail: [balazs.nemeth;attila.lelko;zoltan.antal]@sztaki.hu*
** *School of Music, University of Victoria, 3800 Finnerty Road, Victoria BC V8P 5C2, Canada. E-mail: acsaba@uvic.ca*

Abstract: This paper proposes a collision-free trajectory design for dance choreography of virtual drones using a hierarchical design structure. The trajectory on two levels is designed, i.e., on the level of individual virtual drones and on the level of their centralized coordination. The design on the drone level through reinforcement learning has been carried out, with which dance steps of virtual drones in their motion can be achieved. The coordination level contains a prediction algorithm, which guarantees the avoidance of collision, even at the faultiness of some virtual drones. The effectiveness of the hierarchical design on the example of a Viennese waltz dance choreography in augmented reality has been illustrated.

Keywords: trajectory design; collision avoidance; reinforcement learning; tech&art

1. INTRODUCTION AND MOTIVATION

Motion model formulation and control design for vehicles and mobile robots with high number of constraints on their motion is a challenging task (Murphey and Burdick (2000); Yin et al. (2019)). The control problem of vehicles with special motion becomes more complex in multi-vehicle context, when their coordination for avoiding collision must be guaranteed. This paper deals with one of a special motion, i.e., dancing motion of multiple virtual drones with some predefined rules of Viennese waltz. The desire to visualize aspects or content of music accompanies the performance of music. Historically, practitioners of music visualization have been conductors, dance performers and participants of choreographed ballroom events. The link between music information research and algorithmic/generative visual arts had transferred the process into the digital domain. It has achieved cult status by the first decade of the 2^{nd} Millennia. The motion of drones represents the third generation of music visualization, a re-embodiment that applies to inanimate objects rather than human bodies. A challenge in this field is to avoid fixed scenes in choreography, and thus, decision on dance steps must be performed by virtual drones itself. Learning methods is fruitful for this problem, because it cannot require predefined choreography, and similarly, collision-free trajectory design in the training process of the agent can be incorporated.

Nevertheless, safe learning in the context of unmanned aerial and road vehicles can be a difficult problem, especially under varying physical environment. A general framework for safe learning purposes with certifications on the motion of drones has been presented by Fisac et al. (2019). Another robust control design method for achieving guarantees on performances, if learning-based agents are in the control loop, has been proposed by Németh and Gáspár (2021). Reinforcement learning (RL) with safety guarantees under varying environment, i.e., motion control of automated vehicles in interaction with human-driven vehicles, has also been provided by Németh and Gáspár (2021).

From the viewpoint of dancing feature of drones, some studies have already been proposed. A survey on the topic of using drones for entertainment purposes has been found in the study of Kim et al. (2018). Dancing of real drones with periodical motion in protected area has been proposed by Augugliaro et al. (2013). This platform for implementation of drone dance choreography has been introduced by Schoellig et al. (2010, 2014).

This paper proposes a further solution on the realization of dance choreography, which focuses on virtual drones in augmented reality. Although the proposed trajectory design method can also be used for physical drones, using virtual drones have some advantages. Virtual drones do not pose safety challenges (e.g., risk of causing injury for audience) and thus, protected area for virtual drones is unnecessary. Consequently, the motion of virtual drones can be close to the audience, e.g., over the head of the

⋆ The research has been supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program. The research has been partially supported by the NRDI Office under OTKA Grant Agreement No. K 135512 and by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy".

10.1016/j.ifacol.2022.07.178

audience, which can result in greater experience. Moreover, it does not require high number of physical drones, and the setting of dance show in all areas is time consuming and low-cost.

The contribution of this paper is a design method of collision-free trajectory for dancing drones. As a novelty, the trajectory of virtual drones is partially designed by reinforcement learning, which results that scenes in the choreography are not predefined, only the target positions of the virtual drones must be predefined. The collision-free trajectory is achieved through a hierarchical design structure, which contains RL-based design on the level of drones and a coordination algorithm on a centralized level. An achievement of the paper is that collision-free motion even at faulty operation of some virtual drones can be guaranteed. The effectiveness of the proposed hierarchical design method through a choreography of virtual drones with Viennese waltz steps in augmented reality is demonstrated. Moreover, the fault-tolerance of the proposed hierarchical design through simulation examples is illustrated.

The paper is organized as follows. In Section 2 the trajectory design method is proposed. First, the architecture of the control system is introduced, second, the formulation of motion model and the centralized design level is proposed. Third, the RL-based trajectory design is presented. Section 3 demonstrates the effectiveness of the proposed method through virtual drone dancing performance and finally, in Section 4 the contributions of the paper are concluded.

2. TRAJECTORY DESIGN METHOD FOR DRONE CHOREOGRAPHY

In this section the control design for achieving the requested collision free drone choreography is proposed. First, the architecture of the control system is presented. Second, the formulation of the motion model for learning purposes is shown. Third, the reinforcement learning process on the system is proposed.

2.1 Architecture of the control system

The architecture of the control system, together with its physical implementation is illustrated in Figure 1. The design of collision-free trajectory for n number of virtual drones has two layers.

One of the layer is on the level of drones, i.e., each drone designs its own trajectory on XY plane, independently from the further drones. This process is a RL-based trajectory design, which considers actual XY position, heading angle x_i, y_i, ψ_i and target position $x_{T,i}, y_{T,i}$ in XY plane. Moreover, an additional information for the RL-based design is the requested direction of motion, i.e., forward or backward motion is requested based on the rules of Viennese waltz. The rotation of drones for increment its heading angle in anticlockwise direction is allowed. The drone level design computation is performed at the beginning of each beats. Outputs of this layer are candidate heading angle $\psi_{L,i}$ and candidate velocity $v_{L,i}$.

The role of another layer is the coordination of individual virtual drone trajectories for avoiding collisions. This

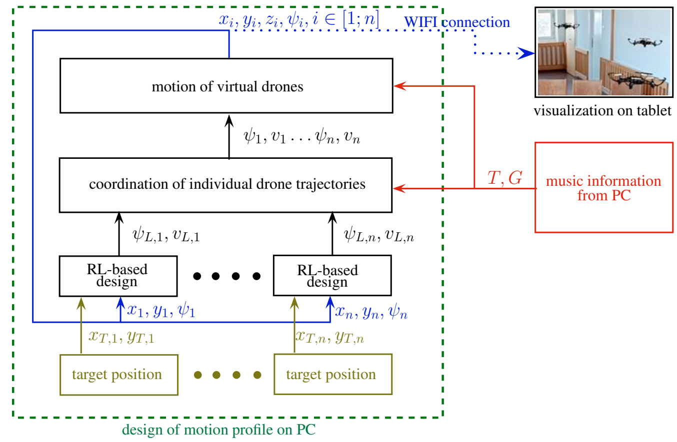


Fig. 1. Architecture and implementation of control system

layer contains a centralized coordination algorithm, which considers $\psi_{L,i}, v_{L,i}$ of all drones and time length of actual beat T. The computation for coordination at the beginning of each beats are performed, similarly to RL-based design layer. The outputs of the coordination layer are heading angles and velocities ψ_i, v_i for each virtual drones. These signals, together with T and the amplitude of music G are used for the generation of the motion profiles, i.e., x_i, y_i, z_i.

The computation of the motion profile on a PC using Matlab is carried out. The computed motion profiles are used for the visualization of virtual drones in augmented reality. Actual motion and heading information through Wi-Fi for a mobile device are transmitted. For the visualization an Android application based on Vuforia engine in Unity is developed. The operation requires a marker (e.g., on the wall) for the positioning of the mobile device.

Information on the music, i.e., time length of beats, amplitude, can be performed in a music analysis software. Among other detection and identification methods for extracting content-based audio features for pre-recorded audio files, beat detection processes are well known and already integrated into various referential audio mixing software. However, real-time extraction of meaningful content-based data from a live audio signal is a novel field and has yet to be solved challenges. Live audio data can be analyzed with Zsa modules in the interactive Max music programming environment (Cipriani and Giri (2019)). The patch can provide beat detection, based on event detection, see Malt and Jourdan (2011). This concept is coded in the Zsa modules, and further contextual data based on spectral music analysis that shapes the motion of virtual drones in alignment with musical predictions. The live data stream of Max analysis is connected to Matlab via UDP protocol with a capacity of direct triggering the motion of drones.

2.2 Formulation of collision-free virtual drone motion

The motion of virtual drones in XY plane is based on the following kinematic model:

$$x_i(k+1) = x_i(k) + T \cdot v_i(k) \sin \psi_i(k), \tag{1a}$$
$$y_i(k+1) = y_i(k) + T \cdot v_i(k) \cos \psi_i(k). \tag{1b}$$

Due to the rules of basic steps in Viennese waltz, the following constraints on the planar motion is formed:

$$\psi_i(k+M) \geq \psi_i(k), \tag{2a}$$
$$sign(v_i(k+M)) = -sign(v_i(k)), \tag{2b}$$

which represent that virtual drones can rotate only in anticlockwise direction, and forward-backward direction of motion is predetermined. In (2) M represents the number of time steps of a beat, and the entire time length of the beat is T.

The collision free motion of virtual drones in XY plane must be guaranteed. The constraint on the motion of virtual drones is formed as

$$\sqrt{(x_i(k) - x_j(k))^2 + (y_i(k) - y_j(k))^2} \geq s_{safe}, \quad \forall i, j \in [1; n], i \neq j, \tag{3}$$

where s_{safe} is safe distance between virtual drones i, j. Constraint (3) in every k time step must be guaranteed, which is the principle of collision-free trajectory design.

Achieving collision free trajectories, i.e., guaranteeing (3), is based on the following algorithm. In the design of safe trajectory, the outputs $\psi_{L,i}, v_{L,i}$ of each RL-based design process are used, which are candidate heading angles and velocities. The acceptability of $\psi_{L,i}, v_{L,i}$ at the beginning of the beats, from the all M points along the beat is checked as follows. The algorithm is performed in every k step, and in every step for each drone i in reverse ordering selection, i.e., $i = n; i = n-1; \ldots i = 1$:

(1) Virtual drone i is selected. The predicted position $x_i(k+1), y_i(k+1)$ using $\psi_{L,i}, v_{L,i}$ is computed.
(2) The sets of critical points in XY plane for virtual drone i are determined. It contains the positions of other drones $x_j(k), y_j(k)$, $\forall j \in \left([1;n]/i\right)$ and static points of the physical area, e.g., walls. Furthermore, the predicted positions $x_h(k+1), y_h(k+1)$ of virtual drones $h \in [i+1; n]$ using $\psi_{L,c}, v_{L,c}$ are computed and they are taken part of the set.
(3) Constraint (3) for drone i and every critical points are checked. If (3) for a critical point is failed, the drone is stopped, i.e., $x_i(k+1) = x_i(k), y_i(k+1) = y_i(k)$.
(4) In the remaining of the beat, drone i is stopped and its stopped position in the determination of the set of critical points for the further drones is incorporated.

The results of the algorithm are collision free trajectories, in which some of virtual drones may be stopped. For achieving aesthetic motion in dance choreography, velocity input v_i for stopped drones is recomputed:

$$v_i = v_{L,i}, \qquad \text{if drone } i \text{ is not stopped}, \tag{4a}$$
$$v_i = v_{L,i}\frac{N}{M}, \quad \text{if drone } i \text{ is stopped at step } k+N, \tag{4b}$$

where $N < M$. (4) guarantees that the motion of all virtual drones are continuous during the beat. In this trajectory design strategy, the heading angle is not modified and thus, $\psi_i = \psi_{L,i}, \forall i = [1; n]$. These values are used for computing the motion of virtual drones.

After collision free trajectories for virtual drones are achieved, the vertical motions of virtual drones are computed. The vertical motion profile as a transformation of an arc between start point A and end point E is formed, see Figure 2. In the generation of $v_i(k)$ the actual amplitude of the music, i.e., volume, has role. Amplitude is characterized through a relative value $G \in [0; 1]$. The higher is the amplitude of the music, the higher is the vertical motion of the virtual drones. The variation of the vertical motion during a beat is expressed as a descent motion between the start and the end points of the actual beat.

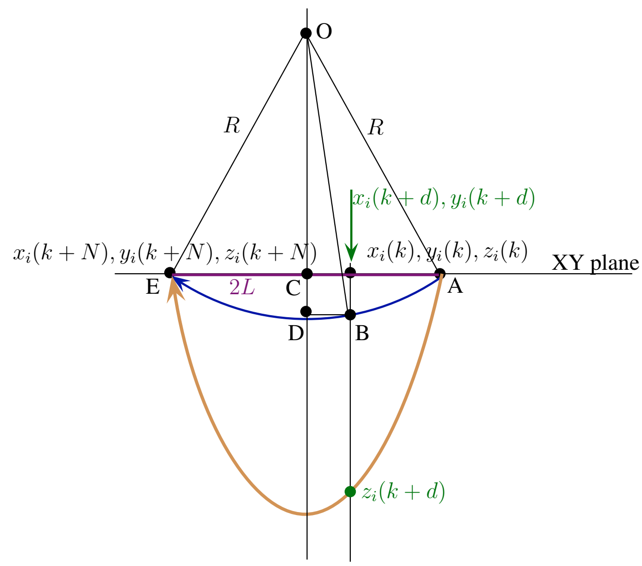


Fig. 2. Illustration of vertical motion profile

The length of the motion during a beat is $2L = v_i \cdot T$, see Figure 2. Radius of the arc R is computed from the triangle OAC, such as

$$R^2 = L^2 + (R - L \cdot G)^2, \tag{5}$$

which means that for $G = 1$ the drone moves along a half-circle, but for $G = 0$ it has straight motion in XY plane. The radius can be computed as

$$R = \frac{L(1 + G^2)}{2G}. \tag{6}$$

Another Pythagorean equation for triangle OBD, for section between A and C can be formed as

$$R^2 = \left(L - 2L\frac{d}{N}\right)^2 + \overline{OD}. \tag{7}$$

Thus, the length of section $\overline{CD}$ is computed as

$$\overline{CD} = \sqrt{R^2 - \left(L - 2L\frac{d}{N}\right)^2} - \sqrt{R^2 - L^2}. \tag{8}$$

The vertical descent motion for section between C and E can be computed similarly.

Finally, the vertical position in $k+d$ step for virtual drone i is computed as

$$z_i(k+d) = z_i(k) - \kappa \cdot \overline{CD}, \tag{9}$$

where $\kappa > 0$ is a ratio for highlight visually the descent motion.

2.3 Trajectory design for individual drones via reinforcement learning

During the dance, the drones must be able to reach target positions individually, and for this reason a neural-network-based trajectory algorithm is designed. The neural network is trained using reinforcement learning, because dancing motion with Viennese waltz basic step in XY plane through constrained equations (2) can be formulated.

The model for describing the motion of the individual virtual drone is the formed kinematic model (1),(2). To imitate dancing motion, virtual drone can have positive yaw rotation angle between 0 and $\frac{\pi}{6}$, and then takes a straight path with a constant chosen percentage of its maximum speed v_{max}. Thus, one dance step consists of a yaw rotation and a linear motion. The dancing motion of a drone consists of the alternations of a forward and a backward step, in case of the former the linear motion is in the direction where the drone is facing (forward motion), and in the case of the latter this direction is the opposite (backward motion).

An example on some steps can be seen in Figure 3. At the beginning, the drone is at a starting position facing towards the positive side of the x axis. The first step is an $\psi(k)$ rotation followed by a forward motion $l(k)$, the second step consists of an $\psi(k+1)$ rotation followed by a backward motion $l(k+1)$, and third step is a forward motion with $\psi(k+2), l(k+2)$.

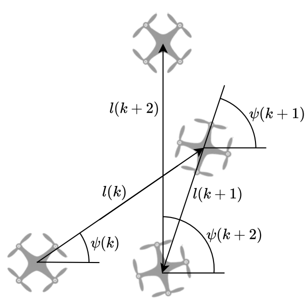


Fig. 3. Illustration of dancing motion with 3 steps

The goal of the control algorithm is for the drone to reach a target position using this step combination. In every step, the control intervention is decided based on the relative position of the goal from the current position of the virtual drone and the information on the direction of the previous step. The action of the network consists of the heading angle ψ and rate of longitudinal velocity $p_i u$, related to the maximum velocity v_{max}. Thus, the motion at the i^{th} step is $v_i = p_i v_{\max}$.

The controller is implemented using reinforcement learning by deep deterministic policy gradient algorithm. The training is carried out using a reward function which is the function of the distance from the goal in each step times minus one, and thus, the reward at the i^{th} step is

$$R_i = -|\mathbf{r}_d - \mathbf{r}_g|, \tag{10}$$

where $\mathbf{r}_d$ is the actual position of the virtual drone and $\mathbf{r}_g$ is the target position. As the result of the training algorithm, the cumulative reward $\sum_{i=1}^{N} R_i$ is maximized.

To achieve a solution with increased generality, the relative target position is normalized through the maximal velocity of the virtual drone. For example, the same neural network can be used in a case of traveling $1m$ distance with a maximum speed of $10cm/s$ or traveling $10m$ distance with a maximum speed of $1m/s$. Consequently, the control interventions, i.e., yaw angle and p_i, result in optimal interventions in both cases.

As a result of this generalization, every distance can be covered within the boundaries of drone motion capability, using similar neural-network-based trajectory design agent. Figure 4 illustrates an example, in which the virtual drone can be navigated anywhere within the grey area with radii of $\underline{r} = 10v_{\max}$ and $\overline{r} = 20v_{\max}$. During the RL-based training process in this paper, the target positions of the virtual drones in this region have been selected randomly in each training episode.

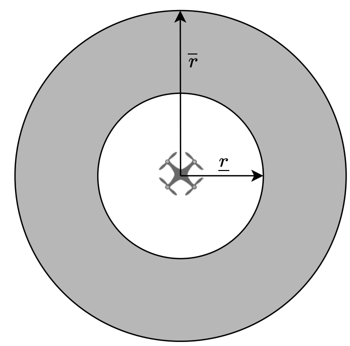
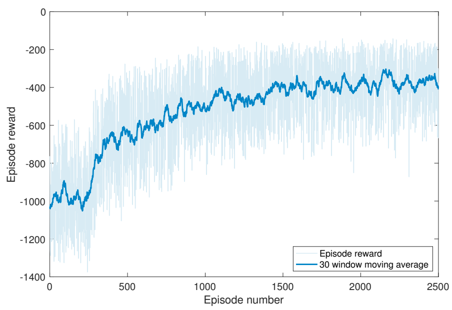


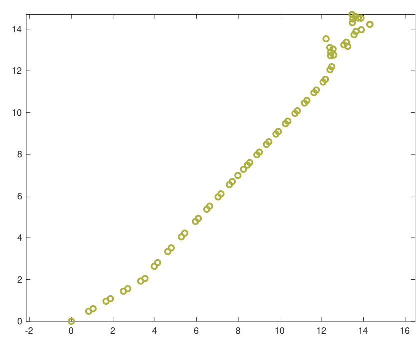
Fig. 4. Illustration of set for selecting target positions

An example on the result of the training can be seen in Figure 5. In this example the structure of the used neural network is selected to have low complexity in order to guarantee the possibility of real-time computation. Both the actor and the critic are feedforward networks with Rectified Linear Unit (ReLU) activation layers. The actor network consists of 3 fully connected layers followed by a scaling layer, which scales the output into the appropriate interval. In the case of the critic network, the action, the observation, and the common paths contain 3 layers. The convergence of the reward function (10) to its maximum value is shown in Figure 5(a). The oscillation of the reward value is resulted by the in-built improving feature of the learning process, i.e., the agent updates the actor and critic using a mini-batch of experiences randomly sampled from a buffer, see Lillicrap et al. (2016). Figure 5(b) shows that the virtual drone drone successfully moves to $20v_{\max}$ distance in this case, i.e., the target positions is $[14.14m\ 14.14m]$. During the simulation the drone turns in the proper direction then takes long forward and short backward steps. Finally, in the virtual drone successfully reaches target position and then, it rotates with dance steps around target.

Remark Training RL-based agent for all drones jointly can be a challenging task. Although it can be advantageous for maximizing N (4), it has some drawbacks. First, a joint design requests fixing the number of virtual drones n. It can limit the applicability of the method with varying number of virtual drones in the dance choreography. Second, training process for joint learning of all virtual drones is also challenging. For example, formulation of reward function for an efficient learning process can be difficult, because in the reward function the virtual drones cannot



(a) Convergence of reward



(b) Trajectory of the virtual drone

Fig. 5. Results on the trained neural network

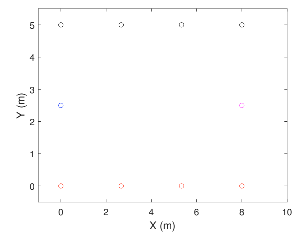
be distinguished. Third, training process for individual virtual drones results in an agent with limited complexity, whose evaluation is a less difficult task.

3. ILLUSTRATIVE EXAMPLE ON THE OPERATION

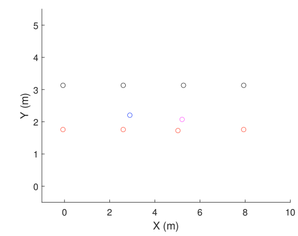
The effectiveness of the proposed method with simulation-based evaluation and with a video on the performance is demonstrated in this section. First, the dancing choreography with 10 virtual drones on a music section of Johann Strauss II: An der schönen, blauen Donau (The Blue Danube) is presented. The illustrative video on the performance can be achieved through `https://youtu.be/0r1-qIIllL8`. Second, the operation of the trajectory design is proposed on two examples, in which the RL-based design layer of some virtual drones are faulty.

In the implementation of the trajectory design, each beat is divided into 10 equidistant sections, i.e., $M = 10$ is selected. Moreover, the rotating motion of virtual drones and their forward/backward motion are separated. On the first three sections of the beat the increase of heading angle is carried out and then, on the rest sections of the beat the dance step with the modification of x_i, y_i, z_i position is carried out.

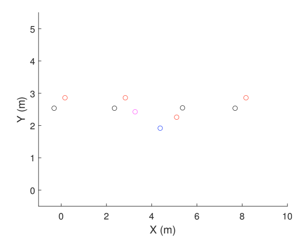
Start and final positions of the virtual drones in XY plane are found in Figure 6. The goal of the choreography is to change positions by the virtual drones facing each other. Consequently, during their motion high number of conflict situations are appeared. For example, virtual drones at $[0m; 2.5m]$ and at $[8m; 2.5m]$ potentially cross the routes of all virtual drones, see Figure 6(a). The positions of the drones at $25s$ and at $38s$ are shown in Figure 6(b,c). It shows that the predefined $s_{safe} = 0.35m$ distance between the virtual drones is kept. The positions of the virtual drones at the end of the music $(67s)$ is found in Figure 6(d). It can be seen that 8 virtual drones have reached their target position. Two drones are slightly farther from their target. Its reason is that their conflict to each other and to the crossing drones request more time.



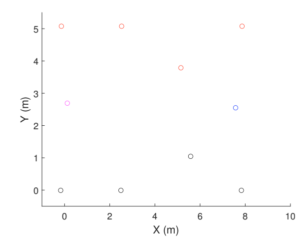
(a) Start positions



(b) Position at $25s$



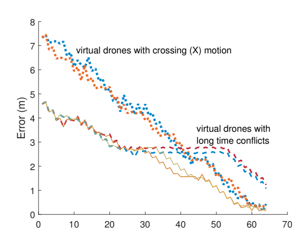
(c) Position at $38s$



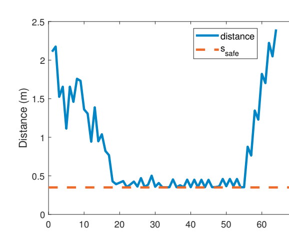
(d) Final positions

Fig. 6. Positions of virtual drones during the choreography

The position errors of the virtual drones from their target are shown in Figure 7(a). It illustrates the effectiveness of the trajectory design, because all of the positions of virtual drones converge to their target. The oscillation in the error signals is resulted by the forward/backward motion of virtual drones, which is the characteristics of Viennese waltz. It can also be seen that two virtual drones, which are slightly farther from their target at the end of the music section, request longer time for solving their conflicts. Moreover, in Figure 7(b) the smallest distance between the virtual drones in each beats is shown. It can be seen that $s_{safe} = 0.35m$ safe distance is kept and thus, a collision-free trajectory is achieved.



(a) Errors from targets

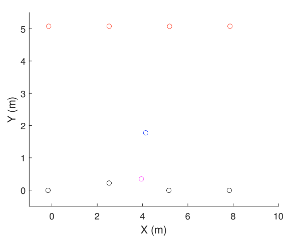


(b) Minimum distance value

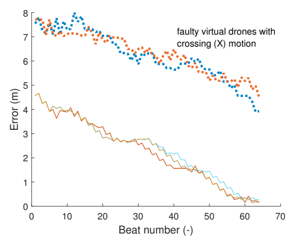
Fig. 7. Positions of virtual drones during the choreography

In the rest of this section two examples are illustrated, which contains virtual drones with faulty RL-based layer on the drone level. In this context faulty means that the outputs of their neural networks differ from their optimal intervention. During the simulation it has been achieved through an addition of a random number to the outputs and thus, the increment of heading angle is increased to $[0; \frac{\pi}{2}]$. Thus, the motion of the faulty virtual drones differs from the healthy drones.

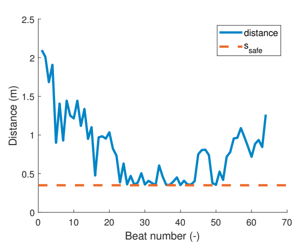
Figure 8 shows a scenario, in which the two virtual drones at $[0m; 2.5m]$, $[8m; 2.5m]$ are faulty. The final positions show that these virtual drones do not reach their targets, see Figure 8(a). Nevertheless, the error values of all drones are reduced, but in case of faulty virtual drones the convergence is slower (Figure 8(b). The effectiveness of the collision-free trajectory design method is illustrated by the characteristics of the minimum distance, see Figure 8(c). In spite of the faulty drones, the safety criterion is not violated and thus, collision during the entire simulation is avoided.



(a) Final positions



(b) Errors from targets



(c) Minimum distance value

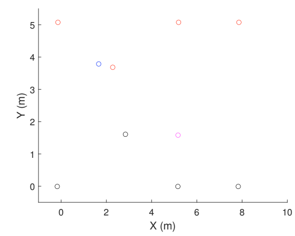
Fig. 8. Scenario with 2 faulty drones

Another simulation with four faulty drones is illustrated in Figure 9. In this scenario two further virtual drones (with starting positions $[2.6m; 0m]$; $[2.6m; 5m]$) are faulty. Similarly to the previous scenario, the faulty virtual drones do not achieve their target position (see Figure 9(a)) and their convergence to the target position is slower, see Figure 9(b). Nevertheless, the collision free trajectory during the entire simulation is guaranteed, see minimum distances in Figure 9(c).

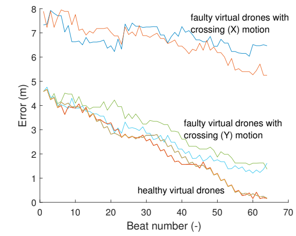
4. CONCLUSIONS

In this paper a collision-free trajectory design method for virtual drones has been proposed, with which dance choreography can be achieved. The simulation results show that the combination of reinforcement-learning-based design and predictive coordination layer is able to guarantee collision free motion, even if some of the virtual drones are faulty. The operation of the trajectory design, together with its implementation in augmented reality, on the example of Viennese waltz dance have been demonstrated.

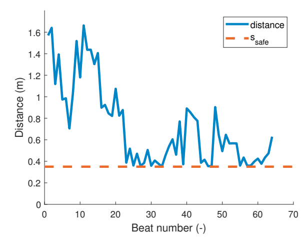
The future challenge of the research is to improve the reduction of conflicts, i.e., create a feedback from the coordination layer to the reinforcement-learning-based trajectory design. Through an advanced coordination the appearance



(a) Final positions



(b) Errors from targets



(c) Minimum distance value

Fig. 9. Scenario with 4 faulty drones

of critical situations can be avoided. From the side of the choreography performance, the detection of beats is planned to connect to the hand motion of conductor, which can provide enhanced experiences for the audience.

REFERENCES

Augugliaro, F., Schoellig, A.P., and D'Andrea, R. (2013). Dance of the flying machines: Methods for designing and executing an aerial dance choreography. *IEEE Robotics Automation Magazine*, 20(4), 96–104.

Cipriani, A. and Giri, M. (2019). *Electronic Music and Sound Design - Theory and Practice with Max 8*, volume 1. Contemponet, 4 edition.

Fisac, J.F., Akametalu, A.K., Zeilinger, M.N., Kaynama, S., Gillula, J., and Tomlin, C.J. (2019). A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7), 2737–2752.

Kim, S.J., Jeong, Y., Park, S., Ryu, K., and Oh, G. (2018). *A Survey of Drone use for Entertainment and AVR (Augmented and Virtual Reality)*, 339–352. Springer Int. Publishing, Cham.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *Int. Conf. on Learning Representations.*

Malt, D.M. and Jourdan, E. (2011). Real-time uses of low level sound descriptors as event detection functions. *Journal of New Music Research*, 40(3), 217–223.

Murphey, T.D. and Burdick, J. (2000). Issues in controllability and motion planning for overconstrained wheeled vehicles. In *Proc. Int. Conf. Math. Theory of Networks and Systems (MTNS).*

Németh, B. and Gáspár, P. (2021). The design of performance guaranteed autonomous vehicle control for optimal motion in unsignalized intersections. *Applied Sciences*, 11(8).

Németh, B. and Gáspár, P. (2021). *Guaranteed Performances for Learning-Based Control Systems Using Robust Control Theory*, 109–142. Springer International Publishing, Cham.

Schoellig, A.P., Augugliaro, F., and D'Andrea, R. (2010). A platform for dance performances with multiple quadrocopters. *Proc. of the IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS), Workshop on Robots and Musical Expressions*, 1–8.

Schoellig, A.P., Siegel, H., Augugliaro, F., and D'Andrea, R. (2014). *So You Think You Can Dance? Rhythmic Flight Performances with Quadrocopters*, 73–105. Springer Int. Publishing, Cham.

Yin, H., Chen, Y.H., and Yu, D. (2019). Vehicle motion control under equality and inequality constraints: a diffeomorphism approach. *Nonlinear Dynamics*, 95(1), 175–194.