



Online planning for human–multi-robot interactive theatrical performance

Ellen A. Cappel¹ · Arjav Desai¹ · Matthew Collins¹ · Nathan Michael¹

Received: 16 February 2017 / Accepted: 16 April 2018 / Published online: 27 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

We propose and evaluate a multi-robot system designed to enable live, improvisational theatrical performance through online interaction between a performer and a robot system. The proposed system translates theatrical performer intent into dynamically feasible trajectories for multi-robot ensembles without requiring prior knowledge of the ordering or timing of the desired robot motions. We allow a user to issue detailed instructions composed of desired motion descriptors in an online setting to specify the motion of varying collectives of robots via a centralized system planner. The centralized planner refines user motion specifications into safe and dynamically feasible trajectories thereby reducing the cognitive burden placed on the performer. We evaluate the system on a team of aerial robots (quadrotors), and show through offline simulation and online performance that the proposed system formulation translates online input into non-colliding dynamically feasible trajectories enabling a fleet of fifteen quadrotors to perform a series of coordinated behaviors in response to improvised direction from a human operator.

Keywords Human–multi-robot interaction · Multi-robot online planning · Multi-robot formation trajectory generation

1 Introduction

We wish to enable a theatrical performer to direct formations of quadrotors online as part of an improvisational perfor-

mance. Although the choreographed motion types are known before the performance, the choice and ordering of commanded actions, their duration, and the timing between them is directed by a storyteller as part of a live interaction; no movement sequences or plans are generated prior to the performance.

We gratefully acknowledge support from ONR Grants N00014-13-1-0821 and N00014-15-1-2929.

This is one of several papers published in *Autonomous Robots* comprising the “Special Issue on Distributed Robotics: From Fundamentals to Applications”.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10514-018-9755-0>) contains supplementary material, which is available to authorized users.

To enable the theatrical performer to provide online choreographic direction to the multi-robot system, the performer’s intent, including instructions specifying lighting, sound, or movement and timing directions, must be translated into individual robot trajectories online. This requirement of real-time intent translation is particularly challenging as the performer can issue a direction at any time and without consideration of the robots’ extents and performance limits, leading to violation of collision and actuator constraints. Additionally, the performer may err in their direction, leading to a motion specification that is logically invalid.

✉ Ellen A. Cappel
eacappel@cmu.edu

Arjav Desai
aadesai@cmu.edu

Matthew Collins
mjcollins@cmu.edu

Nathan Michael
nmichael@cmu.edu

In this work, we detail a novel system to enable a performer to direct a dynamic multi-robot system in an improvisational (or unscripted) manner in an online setting. Our system architecture takes as input theatrical intent, readily specified online by the performer using our proposed input methodology. We employ a centralized planning approach to translate user input into viable motion specifications which respect actua-

¹ The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

tor and collision constraints for all robots in the multi-robot system.

Our proposed framework encodes user intent through predefined descriptors that capture theatric elements such as the light, sound, or spatial-temporal motion of one or multiple robots. The system interprets these descriptors to form *behaviors* composed of formation, motion, and time descriptors (Sect. 3.1). A centralized planner generates behaviors as individual robot trajectories (Sect. 3.2). The system ensures that the commanded behaviors are both logically valid (Sect. 3.3) and dynamically feasible (Sect. 3.4) through motion plan refinement. Specified behaviors that yield unsafe or dynamically infeasible trajectories are refined to yield non-colliding trajectories. These trajectories serve to transition robots from their current states to the desired behavior trajectories with appropriate time-scaling to ensure that the required accelerations fall within actuator limits (Sect. 3.4.1).

The system presented here illustrates a number of design decisions chosen to meet real world challenges inherent in human–multi-robot teaming for theatric applications.

- A parameter-matching based input methodology is used in order to give a performer detailed motion control over formations of robots while still affording the user the ability to specify instructions online. This approach differs slightly from more common human–multi-robot input methodologies, which we review in the Related Works section, which may be widely generalized as requiring only a low-degree of freedom input. The choice of parameters is derived from the narrative basis of the theatric effort, and parameter design choice and human input are detailed in Sects. 3.1 and 3.6 respectively.
- We pursue a centralized planning framework, rather than localized rule-based motion planning at the individual agent level, in order to guarantee motion plan safety. Plan safety is of paramount concern to the theatric application as robot–robot or robot–environment collisions not only lower an audience’s trust in the performance, but can be physically unsafe for the performer or audience in the event that they are located in close proximity to the operating robots.
- Multi-robot motion planning techniques are based on time optimal trajectory generation methods in order to facilitate the theatric narrative. Feasibility verification is performed with respect to kinodynamic constraints, incorporating actuator limits using the differentially flat quadrotor model, and conflicts resolved through robot prioritization and trajectory time scaling. These methods allow well-described motion plan generation reflecting performer theatrical intent. And while overall planning time with a centralized approach will scale with the number of robots, the chosen methodology performs well for the team size required by the application and is respon-

sive to the human operator. (Performer input to the system occurs over the order of multiple seconds as part of the performance, and motion planning, even via an unoptimized Matlab-based implementation, occurs on the order of subseconds).

Prior work has described behavior composition (Cappo et al. 2016) and shape transitions for multi-robot ensembles (Desai et al. 2016). This work contributes a full system architecture composed of the combined prior methodologies. We provide additional discussion of related work, including theatric performance approaches utilizing robotics with a focus on efforts employing multiple aerial vehicles as well as related work in human-robot interfacing and multi-robot control (Sect. 2). We describe our full system approach incorporating prior work (Cappo et al. 2016; Desai et al. 2016) (Sect. 3) and provide extended analysis and experimental results for increased numbers of robots obtained through experiments done in both simulation and on hardware, for up to fifteen robots (Sect. 4). The evaluation includes an illustrative scenario to provide insight into how varied planning conditions resulting from timing changes due to online instruction are accommodated by our approach. We also include extended simulation results to highlight the concept of coverage over the space of behavior inputs, confirm that plans preserve safety and feasibility requirements, and present plan generation timing characteristics. Simulation performance is validated through execution of a randomly chosen selection of 100 behaviors on ten quadrotors in a motion capture arena, and system performance is further highlighted by showing a chosen sequence of complex behaviors executed on fifteen quadrotors.

2 Related work

Two primary challenges must be addressed to enable a performer to direct multi-robot ensembles online. First, a performer must be able to convey artistic intent to the robot system, and second, the system must be able to translate these commands into motion specifications. These requirements are further complicated when accounting for time and computational complexity considerations to both input and execute instructions, as necessitated by the online performance requirement.

In this section, we review how multi-robot performance is addressed in the context of theatrical applications, and specifically with regard to aerial vehicles. We also discuss approaches to human-robot interfacing and multi-robot control, the constituent components of our approach to direct a robotic theatrical performance, with considerations for online use.

2.1 Theatrical applications

Coordinated vehicle deployments within the context of choreographed and improvisational performances are generally characterized as *scripted* or *unscripted*, respectively. Currently, most robot theatric works are scripted, fully specifying all robot trajectories before the performance, including efforts focused on choreographed aerial dance and acrobatic maneuvers (Augugliaro et al. 2013; Lupashin et al. 2014) and light shows including Intel Corporation’s recent performance at the 2017 NFL Super Bowl LI Halftime Show (Intel Corp. 2017) and the Guinness Book of World Record-setting flight of 1000 drones performed in celebration of the 2017 Chinese New Year (Ehang 2017; Huang 2017) by the Ehang Company in Guangzhou, China. If present, human actors in scripted performances (Knight and Gray 2012) such as AURORA’s recent music video (AURORA 2016; The Creators Project 2017) or Cirque Du Soleil’s “Sparked” (Cirque du Soleil 2014; Coxworth 2014) respond to the robots’ motions to invoke a sense or impression of interaction. Alternative strategies seek to blend online operator interaction with scripted robot motions that are predefined by enabling the user to dynamically trigger the start of the motions (Hoffman et al. 2008) as demonstrated by MagicLab and Rhizomatiks Research (2016b, 2016b). However, fully scripted works do not allow for any change to the choreography during the performance, and thus interaction can only be demonstrated by the human performer as a reaction to the robot system. While dynamically triggered motion sequences enable a user to change elements of the performance online, they do not allow the system to respond or adapt to a user’s intent.

2.2 Interpreting human user intent

For robots to move in response to a performer’s intent, a robot system must be able to extract and process motion or task-level information from operator input. The determination of motion requirements from human input is still an active research effort in single-robot domains as well as the emerging field of human–swarm or human–multi-robot interaction. However, the body of literature studying the related questions of what strategy is best used by a human operator to communicate intent to groups of robots, and through what interface methodologies, has grown rapidly in recent years with advances in multi-robot technologies (Kolling et al. 2016). Human operator input for multi-robot systems has been formulated as:

- selection or tele-operation of leaders in leader-follower formulations through GUI (Bashyal and Venayagamoorthy 2008), haptic (Secchi et al. 2015; Setter et al. 2015),

gesture (Stoica et al. 2013), and joystick (Zhou and Schwager 2016) control;

- using a human operator to perform the function of a “switch” to trigger behavior modes in a hybrid-control formulation through GUI (Brown et al. 2014; Leonard et al. 2010) or speech and facial recognition (Pourmehr et al. 2013);
- allowing the operator to control swarm behaviors such as flocking by performing simulated environment modification—mimicking biologically inspired control methods—through managing artificial attraction or repulsion fields in simulation (Kolling et al. 2012), via drawing interfaces (Hauri et al. 2014), or gesture and “demonstration” through motion-mimicking (Alonso-Mora et al. 2015).

These different approaches consider a similar question: what combination of interface and input methodology best maps the reduced dimensionality of a user’s input (a word, gesture, or button-click) to the high dimensional, application- and platform-dependent state-space of multi-robot trajectory generation?

The theatrical application considered in this work presents a significant challenge in balancing the requirements of “easily specified” input with detailed multi-robot motion control. The theatric performance requires a human performer to direct varying groups of robots to portray characters in an episodic narrative. Throughout the narrative, characters (robots) are required to move in and between formations, maintaining constant or varying positional offsets as directed by the performer. The performer also requires the control fidelity to specify travel speeds, flight destinations, and motion patterns.

While tele-operating leaders in a leader-follower formation can accurately guide robots through user-input motion paths, leader designation and tele-operation approaches (Bashyal and Venayagamoorthy 2008; Secchi et al. 2015; Setter et al. 2015; Stoica et al. 2013; Zhou and Schwager 2016) do not easily scale to multi-group scenarios. Conversely, while switching behavior modes (Brown et al. 2014; Leonard et al. 2010; Pourmehr et al. 2013) or performing environment modification (Alonso-Mora et al. 2015; Hauri et al. 2014; Kolling et al. 2012) can be used to direct multi-group applications, these input methodologies may not necessarily yield the high degree of inter-robot coordination desired by our narrative and may be more appropriate for applications which allow individual agents greater flexibility in motion execution. We therefore describe a parameter-matching based input methodology which allows a user to specify multi-robot motions online through a selection of motion *descriptors*, which combine to form a multi-robot *behavior*. Parameter choice, interpretation to multi-robot trajectories, and online specification are described in the following Sects. 3.1, 3.2, and 3.6 respectively.

2.3 Multi-robot trajectory generation and control

The task of generating collision free, dynamically feasible motion plans for all robots once the user's intent is known is itself a highly complex problem subject to many requisite considerations. Depending on application and environment, researchers have examined multi-robot motion-planning under collision (Hoy et al. 2015), communications (Doriya et al. 2015), energy (Häusler et al. 2016; Levy et al. 2014; Mitchell et al. 2016; Riazi et al. 2015), and actuator constraints (Gazi et al. 2015). The planning problem itself is computationally expensive; while certain non-optimal formulations have been shown to be polynomial time in complexity (Röger and Helmert 2012), the problem of finding optimal paths for multi-robot problems is NP-hard (Ratner and Warmuth 1986). Approaches for finding multi-robot motions are therefore numerous, including centralized paradigms using optimization strategies (Augugliaro et al. 2012; Chen et al. 2015; Mellinger et al. 2012) and search-based sampling approaches (Carpin and Pagello 2012; Ferguson et al. 2006); decentralized formulations of both planned and reactive strategies (Alonso-Mora et al. 2016; Amato et al. 2015); and hierarchical planners employing control techniques from across these categories (Sharon et al. 2012; Standley 2010; Wagner and Choset 2015; Zhou and Schwager 2015). This wealth of planning methodologies reflects the diverse nature of multi-robot applications and their respective requirements.

The theatrical nature of the application requires preserving the aesthetic appeal and visual connection between the performer input and team response, both behaviorally and temporally. Aesthetically, it is desirable for robots to move in distinct groups and for group motions to follow a visually obvious structure in order to allow audience members to easily recognize narrative character groups. The performer in the narrative acts as a story-teller, and intersperses their narration to the audience with motion commands to the robots. The performer's instructions to the robots are fully observable by the audience, and to further link performer direction with robot motions, we require robots to transition between behaviors in a time-optimal manner. Therefore, we require a system formulation that can respond to requests in real-time (low latency) with corresponding behaviors that are time-optimal while preserving feasibility and safety.

To meet these requirements, we therefore propose a centralized planning architecture in order to coordinate intra-group robot trajectories to ensure that group-level motion plans reflect the performer's theatrical intent. A centralized planning methodology additionally ensures that all proposed motion plans across all multi-robot teams are collision free in order to maintain performer safety and audience trust in the performance. Motion plan design builds on prior work in multi-robot trajectory generation (Turpin et al. 2013a,b)

in order to determine optimal shape assignments and ensemble motion specifications for user-specified groups of robots. To transition robots in a time optimal manner between desired plans, optimal trajectories are generated by solving an unconstrained quadratic program (Richter et al. 2013a), and trajectory timing refined through online search (Hehn and D'Andrea 2011; Richter et al. 2013a); the differentially flat quadrotor model (Chamseddine et al. 2012) is used to ensure dynamic feasibility given actuator constraints. This proposed formulation seeks to balance low computation times with near-optimal trajectory generation for teams of robots (Desai et al. 2016).

3 System design

To enable human–multi-robot interactive theatrical performance, we propose a full system that provides a methodology for inputting theatrical intent online and translating performer input into dynamically feasible and safe motion plans for teams of robots. We propose a formation-based approach to enable specification of robot team motion in a manner that seeks to reduce the user interaction burden by avoiding individual robot motion specifications. The performer specifies motion descriptors such as formation shape, flight mannerisms, or destination, and the system composes these descriptors into dynamically feasible and safe behaviors. These behaviors then undergo validation and verification checks and if necessary, are modified to meet collision and actuator constraints. The resulting trajectories are distributed to the robot team.

A block diagram of the proposed system, showing user input, behavior generation, validation, and verification and mitigation components is outlined in Fig. 1. In this section, we step through each block pictured in Fig. 1 and describe our approach to specifying user intent (Sect. 3.1); generating multi-robot *behaviors* based on user input (Sect. 3.2); validating (Sect. 3.3) and verifying (Sect. 3.4) behaviors; and mitigating infeasible behaviors by modifying intra-group transitions (Sect. 3.4.1).

3.1 Operator input: *behaviors*

While we do not define a formal grammar, specifying theatrical intent is similar to answering the questions of “who,” “what,” “where,” “when,” and “how.” The user specifies *descriptors*, b_i , that describe which robots a performer intends to direct, what action the robots should take, the target destination, and any characteristic flight mannerisms that the robots should exhibit. Descriptors are composed into an m -length vector called a *behavior*, $\bar{\mathbf{b}}$. Each behavior descriptor, b_i , may take a discrete number of values, and Fig. 2 highlights several behavior descriptors and potential value assignments.

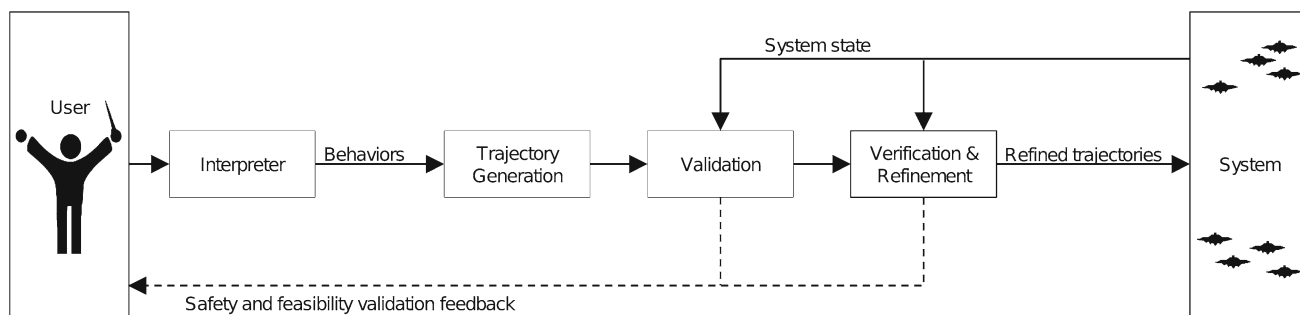


Fig. 1 An overview of the proposed system. User-issued input in the form of *descriptors* is collected by an “Interpreter.” Descriptors, organized into *behaviors*, are output to the trajectory generation subsystem. The proposed multi-robot trajectories exemplifying the user-requested behavior are then checked against the current system state to determine

validity. After passing this check, the proposed trajectories are verified for dynamic feasibility. In the event that the proposed behaviors do not meet feasibility or safety constraints, an online search refines trajectories to satisfy safety and feasibility limits

Denoting B_i as the set of values associated with the behavior descriptor b_i , the total number of potential behaviors achievable by the system is:

$$perm(\bar{\mathbf{b}}) = \prod_{i=1}^m \left(\sum_{k=1}^{K_i} \frac{|B_i|!}{k!(|B_i| - k)!} \right). \tag{1}$$

Equation 1 describes the fact that the total number of potential behaviors achievable by the system is the product of the total number of descriptor combinations across descriptor sets. The total number of descriptor combinations possible for a given descriptor set B_i is given by the summation in Eq. 1, formulated as the sum of binomial coefficients for set B_i . If only a single descriptor may be chosen from a descriptor set, $K_i = 1$. If a combination of descriptors may be chosen (for example, if choosing k robots from the total number of possible robots), K_i may equal up to $|B_i|$.

As a collection of descriptors drawn from each respective set, a motion behavior contains all the requisite information for defining multi-robot trajectories. This section details each descriptor category and explains how descriptors contribute information to the trajectory formulation, such as trajectory duration, endpoint constraints, and motion characteristics.

Behavior duration The starting time of a behavior, t_s , is the time at which the system receives the command from the user.¹ The “time” descriptor specifies the duration of a behavior, giving ending time t_f , the specified timing duration from start time t_s .

Formation specification We describe a formation of robots by specifying each robot’s state in a local reference frame (Desai

$C(t), R(t)$				$S(t)_{xyz}$	$S(t)_{\psi}$	$S(t)$
robots	time	action	goal	shape	heading	manner
[1 ... N]	r seconds	⊙ hover	⊙ v ^{dr}	⊙ as-is	⊙ as-is	⊙ fixed
		↑ takeoff	[Ax, Ay, Az]	□ polygon	x	⋯ variable
		↓ land	[Bx, By, Bz]	— line	y	~ drunk
		→ go-to-target	⋮	⋯ filled	[Ax, Ay]	~ nervous
		↶ forward-rev			[Bx, By]	
		↷ side-side			⋮	
		↕ up-down				
		⊙ circle-target				
		↻ turn-in-place				

Fig. 2 Representative behavior descriptors, b_i , and descriptor sets, B_i , with associated representative values. Descriptor sets are grouped in the table to show the contribution of a descriptor towards an element of the multi-robot trajectory formulation. For example, the “heading” descriptor directly specifies the group trajectory component $S(t)_{\psi}$ as indicated by the column label

et al. 2016), which we call the *shape* frame. The positions and headings of each robot in a local reference frame as a function of time t are $\mathbf{s}(t) = [x(t), y(t), z(t), \psi(t)]^T$, $\mathbf{s} \in \mathbb{R}^3 \times SO(2)$, with a vector $\mathbf{S}(t)$ containing all of the positions in the local reference frame of the n robots in the formation: $\mathbf{S}(t) = [\mathbf{s}_1(t), \dots, \mathbf{s}_n(t)]$. The *shape* descriptor specifies desired starting positions, $\mathbf{s}^{xyz}(t_s)$, in the shape frame (Desai et al. 2016) and the *heading* descriptor specifies $\mathbf{s}^{\psi}(t)$ for each vehicle. A vehicle’s heading is defined relative to its current frame or oriented toward a target in the inertial frame (a theatrical maneuver called “spotting”). Vehicle motions are defined by both the *manner* and *action* descriptors.

Manner The *manner* descriptor is similar to an adjective in language, giving more information about the flight characteristics that each robot should display during the behavior. Two characteristics of interest to the story are “drunk” and “nervous” mannerisms, which a robot performs by mov-

¹ In practice, t_s is set to a value slightly ahead of the instruction receipt time to account for planning computation time, allowing robots to transition between trajectories without discontinuities.

ing along a wobbly course of motion, with slower, larger motions for “drunk” and faster, smaller motions for “nervous.” We represent these motions as bounded polynomial trajectories generated through randomly chosen keypoints obeying timing and distance constraints. Trajectory $\mathbf{s}_n^{xyz}(t)$ for robot n is a spline fit through k keypoints in x , y , and z (Richter et al. 2013b) so that dt_{ij} , the time between each pair of consecutive waypoints i and j , is bounded ($dt_{min} \leq dt_{ij} \leq dt_{max}$) and the sum of all dt 's equals the full time span: $\sum_{i=0, j=i+1}^{i=k-1} dt_{ij} = t_f - t_s$. The position of each keypoint for the n^{th} robot lies within a ball of radius δ centered around the robot's starting position, $\mathbf{s}_n^{xyz}(t_s) \in \mathcal{B}_\delta(\mathbf{s}_n^{xyz}(t_s))$. The bounding values dt_{min} , dt_{max} , and δ are defined on a per-mannerism basis. The “fixed” mannerism denotes regular flight such that the vehicles hold their positions in the local frame throughout the behavior.

All mannerisms, $\mathbf{s}(t)$, must remain within specified limits ensured through appropriate choice of bounding values dt_{min} , dt_{max} , and δ :

$$\mathbf{s}_n^{xyz}(t) \in \mathcal{B}_\delta(\mathbf{s}_n^{xyz}(t_s)), \quad (2)$$

$$|\dot{\mathbf{s}}_n^{xyz}(t)| \leq v_{lim}, \quad (3)$$

$$|\ddot{\mathbf{s}}_n^{xyz}(t)| \leq a_{lim}. \quad (4)$$

Further, the inter-robot clearance distance, d , must be respected at all times, so that for all combinations of robots in $\bar{\mathbf{b}}$:

$$|\mathbf{s}_i^{xyz}(t) - \mathbf{s}_j^{xyz}(t)| \geq d, \quad \forall i, j \in \bar{\mathbf{b}}. \quad (5)$$

In general, we choose to only allow a user to specify a single mannerism descriptor. However, for appropriate choice of bounding values dt_{min} , dt_{max} , and δ , the combinations of mannerisms $\mathbf{s}(t) = \mathbf{s}_1(t) \oplus \dots \oplus \mathbf{s}_j(t)$ will obey the constraints stated in (2)–(5), where \oplus describes a polynomial fit through all keypoints generated for each mannerism \mathbf{s}_i . The vehicles can therefore be “nervous drunks” if required, and the length of the mannerism descriptor set is permitted to be greater than one.

Action The *action* descriptor specifies the motion of the entire formation. Combined with the *goal* descriptor, we can design a trajectory that moves the local formation reference frame through the inertial frame. The state of each robot is defined in the inertial frame at time t by the vector $\mathbf{x}(t)$, containing position coordinates and heading of the vehicle: $\mathbf{x}(t) = [x(t), y(t), z(t), \psi(t)]^T$, $\mathbf{x} \in \mathbb{R}^3 \times SO(2)$. The state of an n vehicle system is given by $\bar{\mathbf{x}}(t) = [\mathbf{x}_1(t), \dots, \mathbf{x}_n(t)]$. We design smooth trajectories for each state-space dimension via time parameterized polynomials up to an appropriate order to ensure smoothness in the trajectories and their derivatives and satisfy dynamic properties of the vehicle control model.

The position of the origin of the local frame with respect to the inertial frame at time t is $\mathbf{C}(t) = [x(t), y(t), z(t)]^T$, $\mathbf{C}(t) \in \mathbb{R}^3$. We denote $\mathbf{R}(t) \in SO(3)$ as the time varying rotation computed from the Euler rotations around the inertial x , y , and z axes, $\mathbf{R}(t) = \mathbf{R}_z(t)\mathbf{R}_y(t)\mathbf{R}_x(t)$. To describe a smoothly varying, differentiable rotation, Euler angles are defined as polynomial trajectories (Mahony et al. 2012).

Actions such as “circle-target” or “turn-in-place” specify formation rotations, while periodic actions (“forward-rev,” “side-side,” and “up-down”) define trajectories along the specified axis through waypoints centered about the target location. All actions are composable with all goals and timing specifications to yield valid polynomial trajectories for $\mathbf{C}(t)$ and $\mathbf{R}(t)$.

Trajectory initial location The starting state of a trajectory governing the motion of a formation of robots is established based on the current states of the robots at the time the instruction is specified. Upon instruction receipt, the local coordinate frame in which the formation shape is defined is established with an identity rotation and located at the mean of the specified robots' current positions and with higher order terms equal to the mean of the robots' higher order states, leading to the definition of states, $\mathbf{C}(t_s)$ and $\mathbf{R}(t_s)$.

Trajectory ending location The ending states, $\mathbf{C}(t_f)$ and $\mathbf{R}(t_f)$, are specified by the “goal” and “action” parameters. Goals are defined as (x, y, z) locations in the inertial frame and actions specify motion primitives in relation to those locations.

3.2 Multi-robot trajectory generation

Behaviors specify all the information required to generate polynomial trajectories for each robot in a formation. As previously described in Sect. 3.1, the descriptors forming a behavior, taken in conjunction with the states of the specified robots at the time the behavior instruction is issued, inform first, the desired start, goal, and any intermediate desired states in the local formation frame (as described by the shape, heading, and manner descriptors); second, the desired start, goal, and any intermediate desired states in the global reference frame (as described by the action and goal descriptors); and third, the desired duration of the behavior. In each respective reference frame, local and global, trajectories may be generated between the desired specified states as polynomial functions of time, enforcing smoothness and continuity constraints to an appropriate order based on the robot dynamic model, using common optimal trajectory generation methods (Mellinger et al. 2012; Richter et al. 2013b). Trajectories for robots in a behavior, $\boldsymbol{\gamma}(t)$, are formed by composing the local shape-frame trajectories, $\mathbf{S}(t)$, with the trajectories describing the motion of the shape frame through the inertial frame,

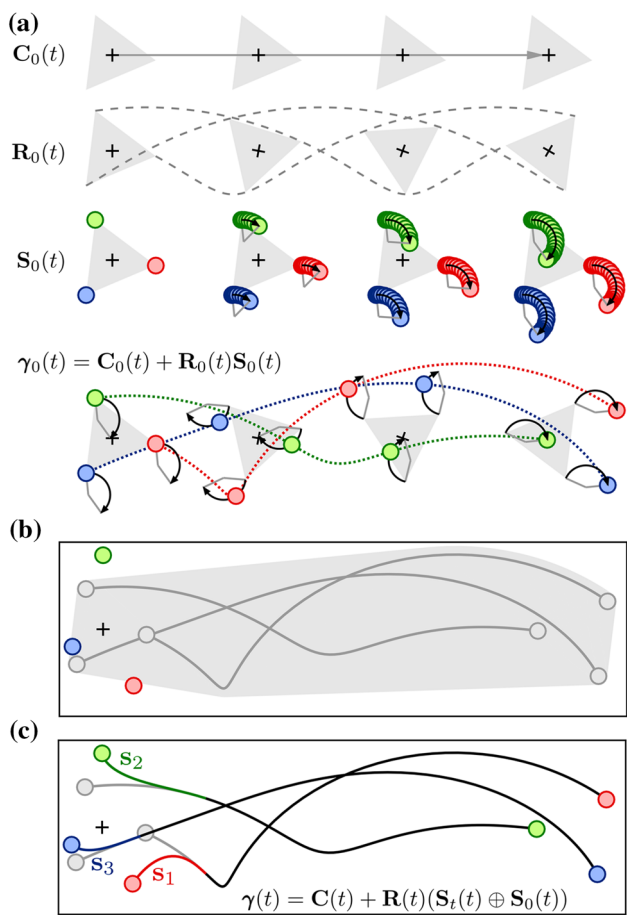


Fig. 3 An illustrated overview of behavior composition, validation, verification, and refinement. Subfigure **a** shows the composition of robot motions in a local shape reference frame with an inertial motion and rotation, illustrating Eq. 6. Subfigure **b** shows validation of the proposed behavior, depicted with respect to the current robot states. Subfigures **c**, **d** show verification given the current robot states and refined behaviors with dynamically feasible transitions, respectively. **a** Proposed behavior, **b** Validation. When started from the mean of the current robot positions, is the convex hull of the proposed trajectory within the flight volume? Does the proposed behaviour violate any user specified rule? **c** Verification. Are the robots able to execute the proposed behavior as specified? **d** $S_i = [S_1, S_2, S_3]$. Can dynamically feasible, collision free transition trajectories be found?

$C(t)$ and $R(t)$, as:

$$\gamma_n(t) = \begin{bmatrix} C(t) + R(t)s_n^{xyz}(t) \\ s_n^\psi(t) \end{bmatrix}, \tag{6}$$

where $s_n(t)$ is one of the n local robot trajectories as specified in $S(t)$, and the superscripts xyz and ψ denote those respective elements of the local state vector. A pictorial representation of an example behavior is shown in Fig. 3a.

3.3 Validation

There are primarily two reasons why a behavior may be invalid in an online setting. First, the current vehicle states may lead to a specified behavior colliding with flight volume boundaries. Second, a user may impose state-transition rules that limit descriptor combinations.

We validate a behavior by first confirming that the descriptors, given the system state, do not result in rule-set violations. An allowable specification is defined as $\{C_0(t), R_0(t), S_0(t)\}$ given the current system state and the descriptor specifications and represents the proposed desired behavior. For example, as shown in Fig. 3b, the convex hull of the behavior is confirmed to remain within the flight volume and is marked as valid.

3.4 Verification and mitigation

Given a valid desired behavior, we verify that the behavior is realizable by checking the following conditions (in order).

1. The current states of the robots specified by the behavior are sufficiently close to the starting states defined by the desired behavior, i.e., $\bar{x}(t_s) \simeq \bar{x}_0(t_s)$.
2. The proposed trajectory accelerations are within the specified limit, $|\ddot{\gamma}(t)| \leq a_{lim}$.
3. The n robots in the behavior maintain an inter-robot spacing greater than or equal to the minimum clearance distance, $|\gamma_i(t) - \gamma_j(t)| \geq d$, for $i, j \in [1, \dots, n]$.

If any condition fails, we immediately proceed to design refined trajectories to mitigate the failure, leading to a dynamically feasible, inter-robot collision-free behavior that remains within the arena volume. We employ the methodology described in Desai et al. (2016) in order to mitigate these conditions, and summarize this approach with respect to our application in the following section.

3.4.1 Behavior transitions

A proposed theatric behavior constructed from user input, as described in Sect. 3.1–3.2, may not meet the three conditions specified at the beginning of this section. We therefore detail a trajectory refinement technique to transition robots from their current states at the time an instruction is issued to the proposed behavior formed as presented in Sect. 3.2. An illustration of this process is shown in Fig. 3c, where the robot’s current states (colored circles) are shown next to the proposed behavior (shown in gray dashed lines), and the solid lines indicate transition trajectories that enable the robots to transition from their current states to the proposed behavior trajectories.

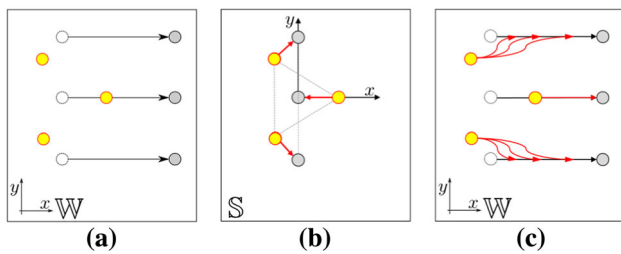


Fig. 4 Overview of the behavior transition process. Figure **a** shows the proposed behavior transition relative to the robots' current states; **b** shows the assignment process in the local *shape* frame; **c** shows how different proposed time scalings for the transition trajectories affect the overall inertial transition trajectories. **a** A behavior transition problem, shown in the inertial frame, \mathbb{W} . Yellow circles are the robots' current states; black arrows from the dashed white circles to the gray circles represent the proposed behavior trajectories, moving robots from left to right in a line formation. **b** Optimal assignment of robots from their current positions (yellow) to the desired line formation, goal positions (gray) in the local shape reference frame, \mathbb{S} . The red arrows show the transition trajectories moving robots from their starting positions to their assigned goal locations. **c** Transition trajectories (red) in the inertial frame, \mathbb{W} . The red arrows depict the same transition trajectory designed in the shape frame performed over different candidate transition times (Color figure online)

We formulate the transition between behaviors as a goal assignment problem. The methodology builds on techniques related to time optimal trajectory generation and feasibility verification with respect to kinodynamic constraints as well as prior work in the areas of multi-robot formation control. We generate optimal trajectories by solving an unconstrained quadratic program to yield an initial trajectory assuming a conservative time-scale (Richter et al. 2013b). The trajectory time-scale is further refined through application of the bisection method to find candidate end times (Hehn and D'Andrea 2011; Richter et al. 2013b) that ensure dynamic feasibility given actuator constraints and the differentially flat quadrotor model (Chamseddine et al. 2012). The proposed formulation seeks to balance low computation times with near-optimal trajectory generation for teams of robots. We also ensure inter-robot collision avoidance by leveraging robot prioritization and trajectory time-scaling to avoid collisions given conflicting trajectories (Turpin et al. 2013b).

Optimal assignment To transition robots from their current states to a proposed behavior, individual robots are first assigned to specific formation positions. We assume that robots are homogeneous and interchangeable with no specific preference to goal locations within a formation and require that the region defined by the convex hull of source and goal locations is obstacle free. Additionally, robot start and goal positions must be located d , a predefined minimum safe distance, apart.

Assignment is performed in the local shape reference frame (as depicted in Fig. 4) and seeks to minimize the associated traversal time costs. The optimal assignment is

computed based on methods detailed in our prior work (Turpin et al. 2013b) and seeks to minimize the p -norm of the costs incurred by the team in order to reach the goal configuration,

$$\phi^* \arg \min_{\phi} = \left(\sum_{i \in I_N} \|P(s_i, g_{\phi_i})\|^p \right)^{\frac{1}{p}}, \quad (7)$$

where I_N is the index set of the robots in the group and s_i and g_{ϕ_i} correspond to the initial and optimally assigned goal configurations of the i^{th} robot, respectively. In this work, we choose to minimize the total distance traveled by the robots and thus let $p = 2$. The optimal assignment is computed via the Hungarian algorithm with $O(N^3)$ computational complexity (Kuhn 2012).

Given the start and goal assignments, we can then use a conservative time duration dt_t to compute transition trajectories $\mathbf{S}_t(t)$ in the local shape frame for the time period from t_s to $t_t = t_s + dt_t$.

Feasible trajectory generation The proposed transition trajectories $\mathbf{S}_t(t)$ are combined with proposed behavior trajectory components $\mathbf{C}_0(t)$ and $\mathbf{R}_0(t)$ over the transition time period t_s to t_t per (6) to yield individual robot trajectories to be executed by the team of robots in order to transition between shapes. However, prior to transmitting the desired trajectory to each robot, we ensure that each trajectory does not require motions that exceed platform actuator constraints. To this end, we compute the maximum mass normalized thrusts required by each trajectory $\boldsymbol{\gamma}_n$ based on the model (Chamseddine et al. 2012) and scale the shape transition trajectory duration, dt_t , accordingly so as to ensure feasibility for all systems.

Alternatively, we note that for visual appeal it is preferable that the robots rapidly transition between shapes. Therefore, if the resulting transition trajectories are overly conservative, we pursue a minimum transition time to enable rapid and feasible shape transitions:

$$\text{minimize: } t_t \quad (8)$$

$$\text{subject to: } -T_{max} \leq \ddot{\boldsymbol{\gamma}}(t) \leq T_{max}, \quad (9)$$

with $t \in [t_s, t_t]$, T_{max} as the maximum allowable mass normalized thrust, and

$$\ddot{\boldsymbol{\gamma}}(t) = \ddot{\mathbf{C}}(t) + \ddot{\mathbf{R}}(t)\mathbf{s}(t) + 2\dot{\mathbf{R}}(t)\dot{\mathbf{s}}(t) + \mathbf{R}(t)\ddot{\mathbf{s}}(t). \quad (10)$$

We solve this minimization problem online via a bisection line search (Cormen et al. 2001), computing the corresponding acceleration time-scale for each candidate time, t_{t_s} , and update the trajectory duration upon termination ($t_t \leftarrow t_{t_s}$).

Online search for minimal-time transitions Given a minimum time transition for all robots, we perform a safety check to ensure that all trajectories preserve a minimum separation distance between robots. If a minimum separation distance is not preserved, prioritization and time-scaling techniques (Turpin et al. 2013b) are applied according to an assigned ordering derived from the robot start and goal positions with respect to the shape specification.

Given the prioritization order, each robot trajectory is checked for collisions against trajectories of higher priority robots. In the event that a collision occurs between robots, the trajectory of the lower prioritized robot is assigned a small, positive time offset to avoid collision with the robot of higher priority. This process is repeated iteratively until no collisions exist between the given robot and all higher priority robots. This prioritization results in collision free trajectories for all robots in the formation relative to the minimum transition time of the highest priority robot.

3.5 Offline verification

System verification is the process of analyzing a system for desired properties, to give evidence that the system meets the desired requirements. One approach to system verification is through formal methods, mathematically based techniques used to reason about systems and their performance (Clarke and Wing 1996; Giammarco and Giles 2018; Kress-Gazit et al. 2018). State of art formal methods for multi-robot applications include approaches such as those based on Linear Temporal Logic (DeCastro et al. 2018; Saha et al. 2014) or satisfiability modulo theory (Saha et al. 2016). These approaches, however, have drawbacks which limit their use for our application. Approaches (DeCastro et al. 2018; Saha et al. 2014, 2016) cite computation times roughly on the order of minutes for problems using two or more robots, which is not conducive to online use or responsive interaction with a human user. While (Saha et al. 2016) presents relaxations which can be used to compute sub-optimal trajectories at faster time scales (problems for ten to twenty robots can be solved on the order of 1–2.5 min), these timescales and the use of discretized motion primitives and sub-optimal trajectories still present drawbacks within the context of our application, where we seek to follow time-optimal trajectories for visual appeal.

In the event that formal verification is not viable, statistical verification through model checking and offline simulation is commonly performed to give quantitative insight into system performance (Clarke and Wing 1996; Giammarco and Giles 2018; Legay et al. 2010). We therefore choose to leverage offline simulation using a high fidelity dynamic simulation environment, simulating all vehicle dynamics including motor response times, across a large number of trials to verify all descriptor combinations assuming a discretization of

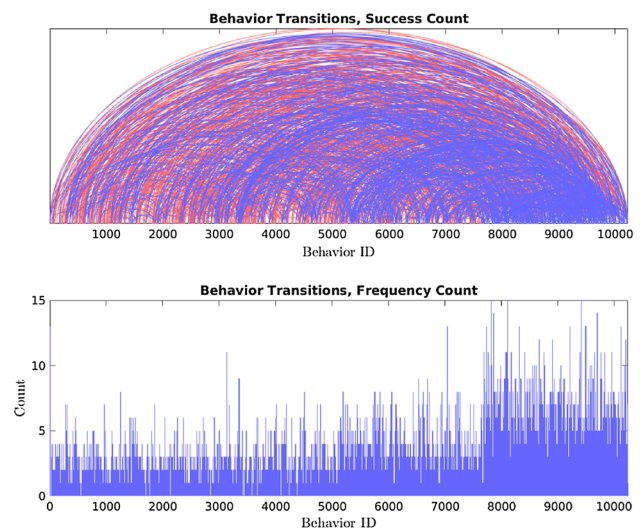


Fig. 5 Plots showing coverage over representative behavior descriptor combinations. Behaviors are validated across varying numbers of robots, with instructions issued at randomly chosen time intervals. Success indicates that the descriptor combination produces a valid behavior and the system is able to interpret, refine, and transform the behavior into a dynamically feasible, collision-free trajectory. Top plot: Arcs describe transition success rates between behaviors, where blue and red correspond to success and failure, respectively. Bottom plot: Behavior validation count. These figures were generated from 48,000 online issued behaviors in simulation, as discussed in Sect. 4.2

the state-space of the system that approximately covers all possible starting and ending states within the flight volume. We depict the results of these offline trials in Fig. 5 and detail both the number of times a descriptor combination is tested and the number of successful behavior transitions. A behavior transition is considered successful if:

1. The descriptor combination forms a valid $\{C, R, S\}$ tuple, meaning the code implementation is error-free;
2. The descriptor combination does not violate a user specified rule; and
3. A dynamically feasible, inter-robot collision-free trajectory is generated from the specified behavior input.

The resulting transition table is employed online to assist in performing fast online validation. Behaviors with intermediate success rates frequently fail due to instruction timing. Therefore, we may choose to use this validation table as a conservative heuristic, and rather than check every online instruction, reject behavior transitions with success rates below a cutoff value.

3.6 Discussion: operator interfacing

We conclude the system description by discussing a performer's interaction with the system, including interface examples and online use.

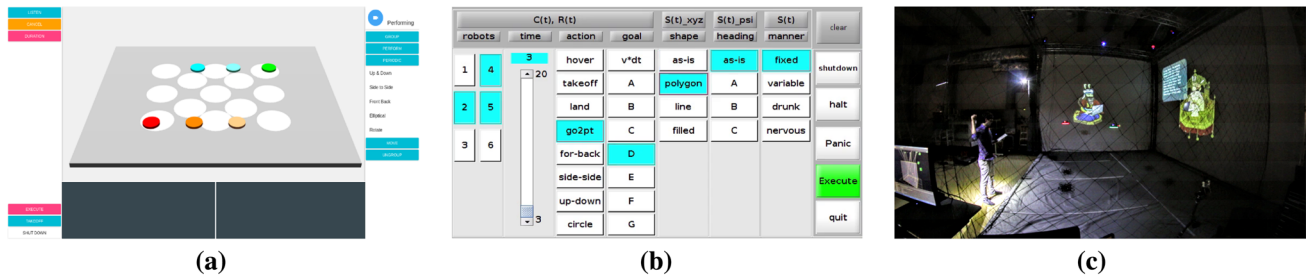


Fig. 6 Interface examples used to convey user-specified descriptors to the multi-robot system. The GUI **a** allows a user to click on simulated robots and select behavior descriptors from organized menus. The GUI **b** shows categorized descriptors as depicted in Fig. 2. Alternatively, a performer can command robots using gestured descriptors, as shown in **c**. In **c**, a motion capture system tracks reflective markers on the per-

former, and this motion data is passed to a gesture recognition system to identify gestured descriptors that are sequentially combined to specify a behavior. **a** A graphical user interface with simulated robots and descriptors as menu items. **b** A graphical user interface for descriptor specification. **c** A performer commands two quadrotors via gestured descriptors

Input space versus ease of use The descriptor based input approach described in this work was implemented after evaluation by the performers in the theatrical application, as performers preferred the balance of ease of use—specifying several descriptors—to the level of afforded motion control. The ability to combine descriptors yields a rich and detailed behavior input space for theatric direction. While the input space is large, descriptor specification was found not to be memory intensive because the descriptor categories chosen to define a behavior paralleled basic language structure. Performers gave commands which were formulated as “You robots (point at robots) [who], go quickly [when] to center stage [where] and spin in place [what] as a circle formation [how].” This “who,” “what,” “where,” “when,” and “how” command specification structure is reflected in the descriptor categories; performers did not struggle to remember descriptors, as they effectively mimicked natural language.

Interfacing The descriptor organizational structure allowed users to easily format behavior commands with little rehearsal, and several interface methodologies were trialed to facilitate online specification. The descriptor-based input methodology is interface agnostic, and the choice of interface was therefore based on the requirements of the theatric application and the performer’s preferences. For example, while implementing a speech-based interface was appealing as a natural extension of the language-based example, a non-vocal input method was pursued so that the performer could, if desired, narrate the story and command the robot system concurrently. Various graphical user interfaces (GUIs) were explored as well as a gesture-based methodology. Two GUIs for descriptor specification are shown in Fig. 6a, b. The GUI format proved useful for system testing and user training, but was not ideal for performance as a performer’s input through screen, keyboard, or mouse was not easily visible to the audience, making it difficult for audience members to clearly

observe communication between the performer and the robot system. The physical modality of a gesture-based framework for specifying descriptors, in contrast, allowed a user’s interactions with the robot system to be a visual part of the performance. A photo of a performer directing robots via gestured descriptors is shown in Fig. 6c. In our implementation, individual gestures corresponded to individual descriptors.

Implementation in practice The physical modality of a gesture-based input was desirable from a theatric perspective, but required greater training time for a performer to learn the gesture notation corresponding to descriptors. Several techniques were therefore employed to reduce training and online specification time. Where possible, gestures were assigned to descriptors based on iconicity (such as making a circular gesture to indicate a circular formation). Additionally, performers chose to allow the system to initialize behaviors with commonly used descriptors as “default” entries. Rather than specifying each descriptor, a performer only needed to specify non-default descriptors (in any order) and send an “execute” gesture to the system. Finally, performers elected to use a combination of both gesture and GUI interfaces to capitalize on the different interface strengths. For example, performers could easily use didactic gestures to select individual or groups of robots (the angle of a user’s pointing gesture was raycast by the gesture recognition processor to select indicated robots). A GUI implemented on a small, touch-based tablet served to show performers their currently specified descriptors (either by gesture or the tablet GUI) and allowed them to modify behaviors before sending. The tablet display also allowed the user to receive more detailed feedback from the system on vehicle or behavior status, in addition to the LED coloration or status patterns at the vehicles themselves.

The input methodology described in this work gives detailed control of robot motions, and descriptor category and structure choices allow intuitive behavior specification.

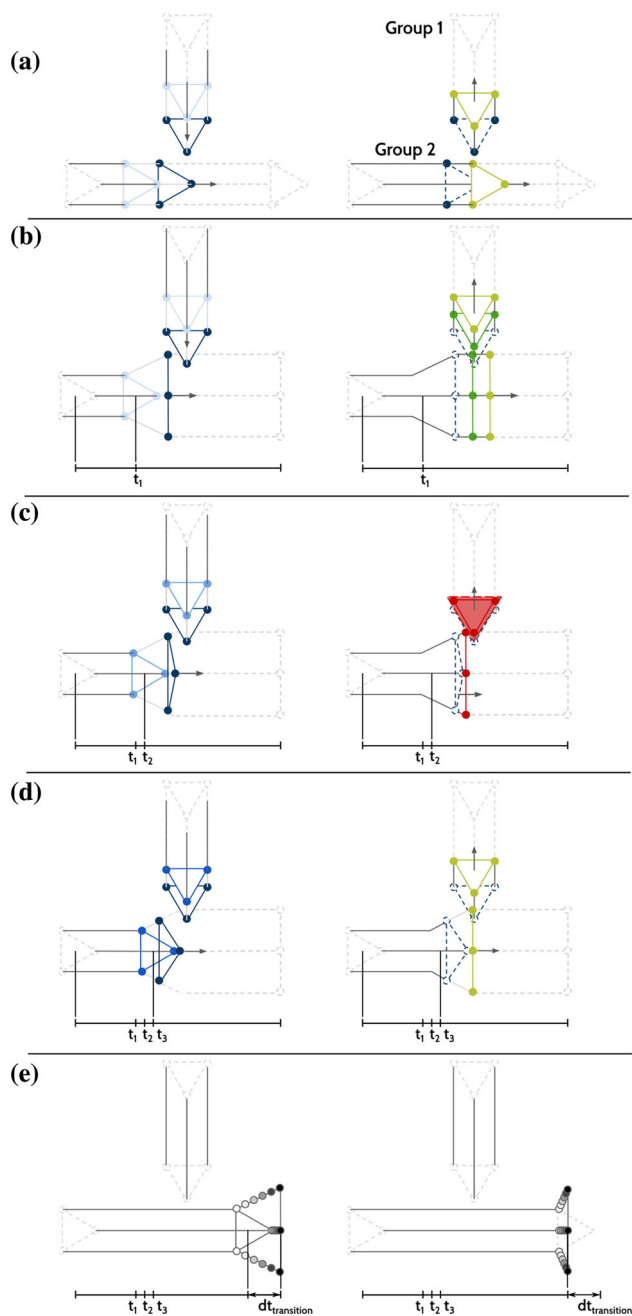


Fig. 7 This series of figures shows planar, timelapsed positions of two group behaviors illustrating the effect of online instruction timing on planning performance. The original, safe motions of two groups are shown in **a**: Group 1 moves vertically down, then up, while Group 2 moves from left to right in a triangle formation. In **b**, issuing a transition instruction with instruction time $t_i \leq t_1$ ensures that Group 2 safely transitions without collisions. In **c**, issuing a transition instruction at $t_i = t_2$ produces a collision with Group 1. In this case, the planner will reject the user's instruction and allow Group 2 to follow its original trajectories. Issuing a transition instruction with $t_i \geq t_3$ is safe again, as shown in **(d)**. In **e**, Group 2 does not have enough time to transition to the asked for line formation and arrive at the destination in the specified amount of time. Consequently, the planner amends the user instruction and extends the instruction time to safely transition the formation. **a** Original group motions, **b** safe transition, **c** colliding transition, **d** safe transition, **e** extended instruction time

The flexibility of the methodology additionally allowed for blending user-preferred input modalities to perform online specification. Design decisions were driven by the specific nature of the theatric application; while interface design is not the focus of this work, this discussion illustrates how the methodology may be adapted given user constraints.

4 Evaluation

We evaluate the proposed approach through both simulation and hardware experiments. We first discuss a brief case study that exemplifies the impact of different online-issued instruction timings on system performance. We then show robustness and coverage of the proposed planner through extensive dynamic simulation, issuing approximately 48,000 randomly generated behaviors. We show that all plans obey safety and feasibility constraints, and we illustrate timing effects as the system scales between 1 and 10 robots. We verify simulation results through hardware experiments, using the CrazyFlie² platform and software framework (Preiss et al. 2016), issuing 100 random behavior instructions online for a 10-robot system. Finally, we perform a sequence of behaviors designed to highlight system features including group splitting, merging, and complex formation changes using 15 quadrotors.

4.1 Case study: impact of instruction timing on system performance

We first explore a simple scenario of two groups moving in close proximity to each other to provide insight into how the timing of a user-issued command impacts the system response. Figure 7 shows a series of illustrations depicting planner responses to a single online instruction, where we change only the time at which the instruction is issued. The original motions of the two groups are shown in Fig. 7a. Group 1 moves vertically down, then up, while Group 2 moves from left to right in a triangle formation. Figure 7b depicts Group 2 safely transitioning to a line formation when the instruction time t_i is less than or equal to the indicated time t_1 on the timeline. However, Fig. 7c highlights that when an instruction is issued at $t_i = t_2$, Group 2 cannot safely transition without a collision. In this case, the planner will reject the user's instruction and Group 2 would follow the original trajectories. Note that if the same instruction to transition is issued at any subsequent time, as shown in Fig. 7d, for $t_i \geq t_3$, the transition can again occur safely. Finally, Fig. 7 shows an example of when a user issues a dynamically infeasible instruction. Group 2 cannot transition from triangle to line and reach the destination in the remaining amount of

² <https://www.bitcraze.io/crazyflie-2/>.

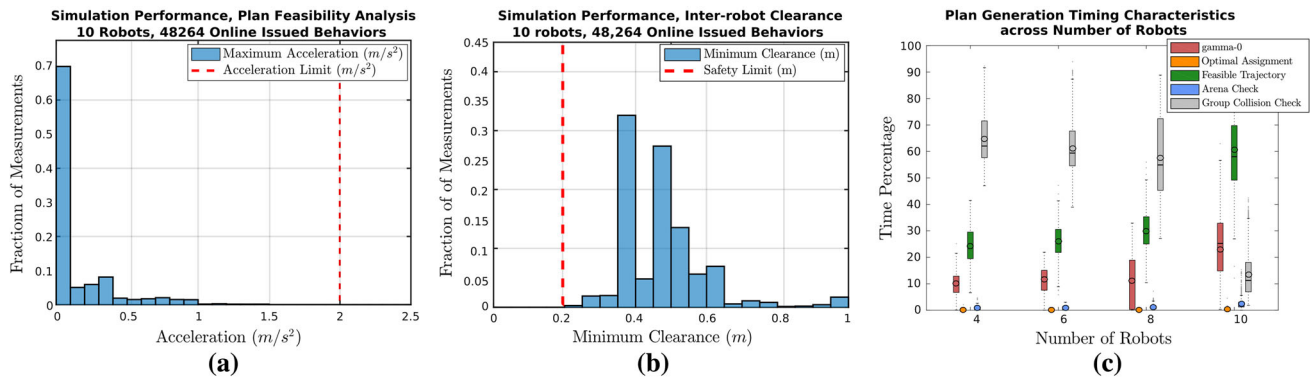


Fig. 8 Dynamic feasibility, inter-robot clearance distance, and timing properties collected over 48,000 online-issued behavior instructions in simulation. Each online issued instruction for a behavior required the computation and execution of a dynamically feasible transition. **a** Dynamic feasibility: Histogram of acceleration measurements, taken every 0.1 s, of the acceleration required by the generated trajectories for all robots over 48,000 online issued instructions. All accelerations

are below the specified acceleration limit **b** Safety: Histogram of robot–robot distance measurements, taken every 0.1 s, across all robots over 48,000 online issued instructions. All inter-robot clearance distances are above the specified safety threshold **c** Timing characterization of various stages of the planning strategy and scaling properties given increasing robot numbers

time. However, the planner can amend the user instruction and extend the instruction time to safely transition the formation.

4.2 Robustness and coverage

We perform evaluation in a high fidelity dynamic simulation environment, simulating all vehicle dynamics including motor response times, to show the robustness of our approach and the associated coverage over the space of behaviors. All behavior instructions were issued at random times during the course of currently executing behaviors. This required the system to generate dynamically feasible and safe transition trajectories given the (randomly chosen) current system state, or recognize that the transition given the current system state was infeasible. Figure 5, as described in Sect. 3.5, shows the coverage results over 48,000 randomly issued behaviors. This plot reports the number of times a behavior is generated as well as the success rate of the behavior transition.

Figure 8 shows that the planner always generates dynamically feasible and safe plans for valid behavior transitions. All motion plan accelerations remain below the specified limit (Fig. 8a), and all plans maintain safe inter-robot clearances (Fig. 8b). We further report the scaling properties of the methodology with increasing numbers of robots in Fig. 8c, showing how portions of the approach scale, as percentages of total computation time, with increasing numbers of robots. While the computation time required by a centralized approach will increase as behaviors are planned for additional numbers of robots, the presented methodology is well able to handle the numbers of robots required by the theatric presentation even when run as an unoptimized MATLAB implementation.

4.3 Hardware experiments

Full evaluation over all simulated behaviors is not feasible in hardware. We therefore verify simulation results by running 100 random behavior transitions in hardware on 10 quadrotors. Each individual behavior was performed for a random amount of time typically lasting from between 30 s to 1 min in length, before a currently executing behavior was interrupted with a new behavior instruction. The Crazyflie platforms were used to validate the methodology because due to their small size, a team of ten vehicles was able to execute multi-group behaviors in a limited motion capture volume. We note that the flight time of the Crazyflie robot platforms, however, was between only 4 and 6 minutes given the energy drain of the LEDs on the batteries and the aggressiveness of chosen behaviors. To validate 100 behavior transitions over a 10-robot team, we therefore flew the team in over ten trials, performing over 100 battery changes across all vehicles, for a total in-air flight time of approximately 1 hour.

A photo of all ten quadrotors in formation, performing a representative behavior from the trials, is shown in Fig. 9c. Figure 9a, b present the accelerations and minimum clearance distances exhibited by all quadrotors over the full flight time composed of all trials. These hardware results verify the simulation experiments. We observe that the reported vehicle accelerations remained under our planning acceleration limit, and all clearance distances between vehicles remained within the stated limits. Hardware experimental data are consistent with simulation results and confirm our dynamic simulation trials.

While the 100 behavior transition results reported in Fig. 9a, b were randomly chosen to cover the behavior input space, a final demonstration of planner capability via system

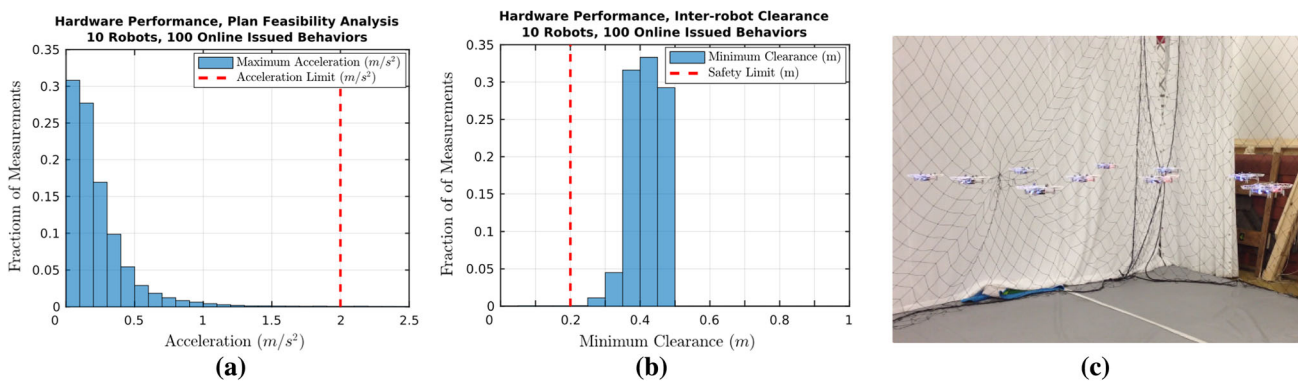


Fig. 9 Dynamic feasibility **a** and inter-robot clearance distances **b** measured over the course of 100 online-issued behavior instructions (each requiring the computation and execution of a dynamically feasible transition) to 10 quadrotors flying in a motion capture arena **(c)**. **a** Dynamic feasibility: Histogram of acceleration measurements, taken every 0.1 s, of the measured acceleration exhibited by 10 quadrotors over 100 online

issued instructions. All accelerations are below the specified acceleration limit. **b** Safety: Histogram of measured robot–robot distances, taken every 0.1 s, across 10 quadrotors responding to 100 online issued instructions. All inter-robot clearance distances are above the specified safety threshold. **c** Image of 10 quadrotors flying in formation during a sequence of 100 online-issued behaviors

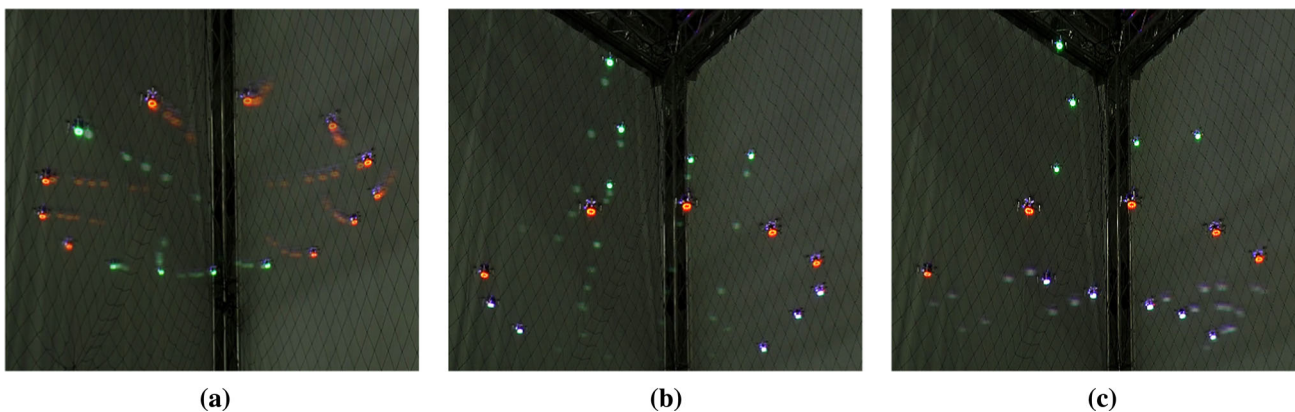


Fig. 10 Online behavior execution for multi-group shape formation for a fifteen quadrotor ensemble. **a** Two groups of quadrotors (green and red) merge to form a circular formation of fifteen quadrotors. **b** A group of 5 quadrotors (green) leave a circle formation and move to a new loca-

tion as a triangle formation. **c** A further group of 5 quadrotors (blue) split from their previous group and merge to form a line formation (Color figure online)



Fig. 11 Photos from a theateric performance. Clockwise, from top left: User practicing gesture-based input; two groups forming lines; one quadrotor performing a solo; red team transitions across from the blue

team; bottom row: two groups of three quadrotors in triangle formations circle each other by performing a rotation maneuver as a formation of six

performance was performed using fifteen quadrotor platforms. For this performance, a set of complex behaviors was chosen which included splitting and merging of groups and performing periodic maneuvers with varied flight formations. These instructions were issued online to the fifteen robot fleet to demonstrate the scalability of the approach and highlight the system's performance abilities. This performance is shown in an accompanying video, available online.³

Additionally, timelapsed images of representative multi-group merging and splitting choreographies performed by the 15-robot team are shown in Fig. 10. We additionally show images from a user-directed theatric performance (Fig. 11), where a performer employed the described system to direct varying numbers of robots in a three act narrative.

5 Conclusion and future work

In this work, we develop a system to enable multi-robot trajectory generation in an online context specifically for an improvisational theatric application, using motion descriptors to allow a performer to specify theatric intent and direct robot choreographies online and using time-aware trajectory formulation methods for validation, verification, and trajectory refinement to systematically generate dynamically feasible and collision free motions. We show through evaluation that the proposed system design yields a robust approach capable of enabling online theatrical performances.

In the future, we will extend the system to incorporate learning techniques to improve system performance and user interaction over many training and performance sessions. These extensions include the generation of new behaviors building on past examples and creating “macros” of recurrent behavior combinations in order to reduce performer command repetition. Further, we are investigating modeling users over repeated interactions in order to anticipate user instructions (i.e., behavior input auto-completion) with the goal of reducing latency by preemptively validating and refining potential motions.

References

- Alonso-Mora, J., Lohaus, S.H., Leemann, P., Siegart, R., & Beardsley, P. (2015). Gesture based human–multi-robot swarm interaction and its application to an interactive display. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 5948–5953).
- Alonso-Mora, J., Montijano, E., Schwager, M., & Rus, D. (2016). Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 5356–5363).
- Amato, C., Konidaris, G., Cruz, G., Maynor, C. A., How, J. P., & Kaelbling, L. P. (2015). Planning for decentralized control of multiple robots under uncertainty. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1241–1248). IEEE.
- Augugliaro, F., Schoellig, A. P., & D'Andrea, R. (2012). Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1917–1922).
- Augugliaro, F., Schoellig, A. P., & D'Andrea, R. (2013). Dance of the flying machines: Methods for designing and executing an aerial dance choreography. *IEEE Robotics and Automation Magazine*, 20(4), 96–104.
- AURORA, YouTube Music, & Carnegie Mellon University. (2016). AURORA: Winter Bird https://www.youtube.com/watch?v=PYdm_Z1wnQ0. Accessed 31 January 2017.
- Bashyal, S., & Venayagamoorthy, G. K. (2008). Human swarm interaction for radiation source search and localization. In *IEEE swarm intelligence symposium* (pp. 1–8). IEEE.
- Brown, D. S., Kerman, S. C., & Goodrich, M. A. (2014). Human–swarm interactions based on managing attractors. In *Proceedings of the ACM/IEEE international conference on human–robot interaction* (pp. 90–97). Bielefeld, Germany .
- Cappo, E. A., Desai, A., & Michael, N. (2016). Robust coordinated aerial deployments for theatrical applications given online user interaction via behavior composition: Distributed. *Robotics and Autonomous Systems*, 27, 171.
- Carpin, S., & Pagello, E. (2012). On parallel RRTs for multi-robot systems. In *Proceedings of the 8th conference on Italian association for artificial intelligence* (pp. 834–841).
- Chamseddine, A., Zhang, Y., Rabbath, C. A., Join, C., & Theilliol, D. (2012). Flatness based trajectory planning/replanning for a quadrotor aerial vehicle. In *IEEE transactions on aerospace and electronic systems* (pp. 2832–2848).
- Chen, Y., Cutler, M., & How, J. P. (2015). Decoupled multiagent path planning via incremental sequential convex programming. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 5954–5961). IEEE, Seattle, WA.
- Cirque du Soleil, ETH Zurich, and Verity Studios. Sparked: A live interaction between humans and quadcopters (2014). <http://flyingmachinearena.org/sparked/>. Accessed 14 July 2016.
- Clarke, E. M., & Wing, J. M. (1996). Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4), 626–643.
- Cormen, T. H., Stein, C., Rivest, R. L., & Leiserson, C. E. (2001). *Introduction to algorithms* (2nd ed.). New York: McGraw-Hill Higher Education.
- Coxworth, B. (2014). Cirque du Soleil and ETH Zurich collaborate on human/drone performance. Gizmag. <http://www.gizmag.com/cirque-du-soleil-sparked-drone-video/33921/>. Accessed 14 July 2016.
- DeCastro, J. A., Alonso-Mora, J., Raman, V., Rus, D., & Kress-Gazit, H. (2018). Collision-free reactive mission and motion planning for multi-robot systems. In *Robotics research* (pp. 459–476). Springer.
- Desai, A., Cappo, E. A., & Michael, N. (2016). Dynamically feasible and safe shape transitions for teams of aerial robots. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 5489–5494). Daejeon, Korea.
- Doriya, R., Mishra, S., & Gupta, S. (2015). A brief survey and analysis of multi-robot communication and coordination. In *2015 International conference on computing, communication and automation (ICCCA)* (pp. 1014–1021). IEEE .
- Ehang. (2017). Chinese new year 2017 lantern festival. <http://www.ehang.com/cn/news/228.html>. Accessed 15 February 2017.

³ <http://www.andrew.cmu.edu/user/ecappo/AURO17.mp4>.

- Ferguson, D., Kalra, N., & Stentz, A. (2006). Replanning with RRTs. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1243–1248). IEEE.
- Gazi, V., Fidan, B., Marques, L., & Ordóñez, R. (2015). Robot swarms: Dynamics and control. In *Mobile robots for dynamic environments* (pp. 81–85). ASME.
- Giammarco, K., & Giles, K. (2018). Verification and validation of behavior models using lightweight formal methods. In *Disciplinary convergence in systems engineering research* (pp. 431–447). Springer.
- Hauri, S., Alonso-Mora, J., Breitenmoser, A., Siegwart, R., & Beardsley, P. (2014). Multi-robot formation control via a real-time drawing interface. In *Field and service robotics* (pp. 175–189). Springer.
- Häusler, A. J., Saccon, A., Aguiar, A. P., Hauser, J., & Pascoal, A. M. (2016). Energy-optimal motion planning for multiple robotic vehicles with collision avoidance. *IEEE Transactions on Control Systems Technology*, 24(3), 867–883.
- Hehn, M., & D'Andrea, R. (2011). Quadcopter trajectory generation and control. In *IFAC world congress* (pp. 1485–1491).
- Hoffman, G., Kubat, R., & Breazeal, C. (2008). A hybrid control system for puppeteering a live robotic stage actor. In *RO-MAN 17th IEEE international symposium on robot and human interactive communication* (pp. 354–359). IEEE.
- Hoy, M., Matveev, A. S., & Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica*, 33(03), 463–497.
- Huang, E. (2017). Watch a record-setting 1000 drones take to the sky in China to celebrate the lantern festival. <https://qz.com/908927/watch-a-guinness-record-setting-1000-drone-show-in-guangzhou-china-celebrating-lantern-festival-and-end-of-chinese-new-year/>. Accessed 15 February 2017.
- Intel Corp., Lady Gaga, Pepsi, National Football League. (2017). Pepsi zero sugar super bowl II halftime show. <http://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html>. Accessed 15 February 2017.
- Knight, H., & Gray, M. (2012). Acting lesson with robot: Emotional gestures. In *Proceedings of the ACM/IEEE international conference on human–robot interaction* (pp. 407–407). IEEE.
- Kolling, A., Nunnally, S., & Lewis, M. (2012). Towards human control of robot swarms. In *Proceedings of the ACM/IEEE international conference on human–robot interaction* (pp. 89–96). ACM.
- Kolling, A., Walker, P., Chakraborty, N., Sycara, K., & Lewis, M. (2016). Human interaction with robot swarms: A survey. *IEEE Transactions on Human–Machine Systems*, 46(1), 9–26.
- Kress-Gazit, H., Lahijanian, M., & Raman, V. (2018). Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 18, 23.
- Kuhn, H. (2012). The hungarian method for assignment problem. In *Naval research logistics quarterly* (pp. 83–97).
- Legay, A., Delahaye, B., & Bensalem, S. (2010). Statistical model checking: An overview. In *International conference on runtime verification* (pp. 122–135). Springer.
- Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., & Zhang, F. (2010). Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, 27(6), 718–740.
- Levy, D., Sundar, K., & Rathinam, S. (2014). Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2014/131450>
- Lupashin, S., Hehn, M., Mueller, M. W., Schoellig, A. P., Sherback, M., & D'Andrea, R. (2014). A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics*, 24(1), 41–54.
- MagicLab, Rhizomatiks Research: DroneMagic—Behind the scenes. (2016a). <https://www.youtube.com/watch?v=JEWXBEDAq60>. Accessed 14 June 2016.
- MagicLab, Rhizomatiks Research: MagicLab 24 drone flight. (2016b). http://www.magiclab.nyc/research/drone_magic/. Accessed 14 June 2016.
- Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation Magazine*, 19(3), 20–32.
- Mellinger, D., Kushleyev, A., & Kumar, V. (2012). Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 477–483). Saint Paul, MN.
- Mitchell, D., Cappel, E. A., & Michael, N. (2016). Persistent robot formation flight via online substitution. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 4810–4815). IEEE, Daejeon, Korea.
- Pourmehr, S., Monajjemi, V. M., & Vaughan, R., Mori, G. (2013). “You two! Take off!”: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 137–142). IEEE.
- Preiss, J. A., Hönig, W., Sukhatme, G. S., & Ayanian, N. (2016). Crazyswarm: A large nano-quadcopter swarm
- Ratner, D., & Warmuth, M. K. (1986). Finding a shortest solution for the $N \times N$ extension of the 15-PUZZLE is intractable. In *AAAI* (pp. 168–172).
- Riazi, S., Bengtsson, K., Wigström, O., Vidarsson, E., & Lennartson, B. (2015). Energy optimization of multi-robot systems. In *2015 IEEE International conference on automation science and engineering (CASE)* (pp. 1345–1350). IEEE.
- Richter, C., Bry, A., & Roy, N. (2013). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Proceedings of the IEEE international conference on robotics and automation*.
- Richter, C., Bry, A., & Roy, N. (2013). Polynomial trajectory planning for quadrotor flight. In *Proceedings of the Robotics: Science and System*.
- Röger, G., & Helmert, M. (2012). Non-optimal multi-agent pathfinding is solved (since 1984). In *SOCs*.
- Saha, I., Ramaithitima, R., Kumar, V., Pappas, G. J., & Seshia, S. A. (2014). Automated composition of motion primitives for multi-robot systems from safe LTL specifications. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1525–1532). IEEE.
- Saha, I., Ramaithitima, R., Kumar, V., Pappas, G. J., & Seshia, S. A. (2016). Implan: Scalable incremental motion planning for multi-robot systems. In *ACM/IEEE 7th International conference on cyber-physical system* (pp. 1–10). IEEE.
- Secchi, C., Sabattini, L., & Fantuzzi, C. (2015). Conducting multi-robot systems: Gestures for the passive teleoperation of multiple slaves. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2803–2808). IEEE.
- Setter, T., Fouraker, A., Kawashima, H., & Egerstedt, M. (2015). Haptic interactions with multi-robot swarms using manipulability. *Journal of Human–Robot Interaction*, 4(1), 60–74.
- Sharon, G., Stern, R., Felner, A., & Sturtevant, N. R. (2012). Meta-agent conflict-based search for optimal multi-agent path finding. *SOCs*, 1, 39–40.
- Standley, T. S. (2010). Finding optimal solutions to cooperative pathfinding problems. *AAAI*, 1, 28–29.
- Stoica, A., Theodoridis, T., Hu, H., McDonald-Maier, K., & Barrero, D. F. (2013). Towards human-friendly efficient control of multi-robot teams. In *International conference on collaboration technologies and systems* (pp. 226–231). IEEE.

The Creators Project: Behind the scenes of a flying robot-built cocoon. (2016). <https://youtu.be/xcVoS6DpFf8>. Accessed 31 January 2017.

Turpin, M., Michael, N., & Kumar, V. (2013). Concurrent assignment and planning of trajectories for large teams of interchangeable robots. In *Proceedings of the IEEE international conference on robotics and automation*

Turpin, M., Mohta, K., Michael, N., & Kumar, V. (2013). Goal assignment and trajectory planning for large teams of aerial robots. In *Proceedings of the robotics: science and system*

Wagner, G., & Choset, H. (2015). Subdimensional expansion for multi-robot path planning. *Artificial Intelligence*, 219, 1–24.

Zhou, D., & Schwager, M. (2015). Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1737–1742). Seattle, WA.

Zhou, D., & Schwager, M. (2016). Assistive collision avoidance for quadrotor swarm teleoperation. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1249–1254). Stockholm, Sweden.



Ellen A. Cappo is a Ph.D. student in the Robotics Institute at Carnegie Mellon University in Pittsburgh, Pennsylvania, working with Nathan Michael. She received her S.B. degree in Mechanical Engineering from the Massachusetts Institute of Technology in 2007 and her S.M. from the Robotics Institute at Carnegie Mellon University 2014. Her research focuses on online planning and control of multi-robot systems.



Arjav Desai is a research associate at the Robotics Institute, Carnegie Mellon University. He works on multi-robot planning, coordination, and control.



Matthew Collins is a research associate at the Robotics Institute at Carnegie Mellon University. He works on hardware and controls for multi-aerial vehicles.



Nathan Michael is an Assistant Research Professor in the Robotics Institute at Carnegie Mellon University in Pittsburgh, Pennsylvania. Previously, he received a Ph.D. from the Department of Mechanical Engineering at the University of Pennsylvania (2008) before transitioning into a Research Faculty position in the same department (2010). His research focuses on enabling autonomous ground and aerial vehicles to robustly operate in uncertain environments with emphasis on robust state estimation, control, and cooperative autonomy.