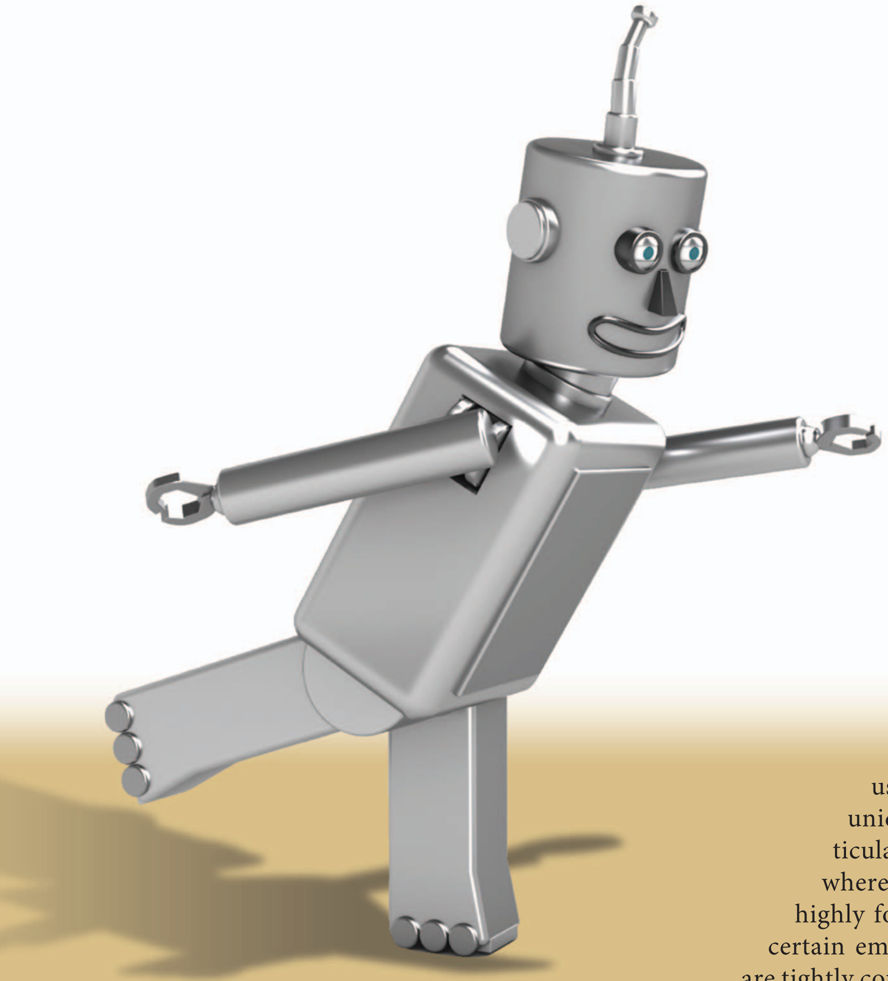
Automatic Sequencing of Ballet Poses

A Formal Approach to Phrase Generation



By Amy LaViers, Yushan Chen, Calin Belta, and Magnus Egerstedt

Understanding human behavior is a goal in many disciplines: behavioral psychology, neuroscience, robotics, computer vision, artificial intelligence, athletics, and dance, to name a few. Thus, this article may be understood in terms of a larger dialogue about how to represent human movement with a concise parameterization. Formal methods for verification can play a surprising role in this discourse. We will ensure that our system evolves according to specific style-based guidelines in a framework where verification methods become an integral part of the system model itself.

Our system model describes human movement, and our specifications control this model, preventing it from entering disallowed and undesirable states. This allows us to produce sequences of movements in a unique style with particular emotive effects. Particularly, we draw inspiration from classical ballet where the set of allowable movements is sparse and highly formalized and where the particular style and certain emotive interpretation of movement sequences are tightly coupled.

Such a framework should cascade into the field of humanoid robotics. The *emotive content* found in the formalized movements like those in dance is also present in the casual, pedestrian gestures of everyday body language; furthermore, the high-level instructions required to generate such a set of movements are a natural augmentation to basic robotic task specification. An abstraction based on such principles could one day allow humanoid robots to operate in dynamic environments amid social

PHOTO COURTESY OF WWW.ISTOCKPHOTO.COM/ ALEXANDR MITIUC

Digital Object Identifier 10.1109/MRA.2011.942118
Date of publication: 13 September 2011

human beings where movement style facilitates communication [9] and where instructions are given at a high level.

A number of human motions have been successfully encoded using dynamic motion primitives [4] and labeled with tasks, such as reaching, drawing, and arguably walking. These primitives or movemes [2] are designed to produce rich and complex human-like motions through systematic, temporal composition. Traditionally, these primitives are obtained from empirical data, e.g., collected using motion capturing devices, that is segmented (often by hand [1] but with progress toward automatic segmentation [6]) into appropriate chunks and stored in a motion library [16]. But full-fledged behaviors do not follow predictable, continuous trajectories as humans constantly make discrete decisions and may abruptly change behavioral modes, and there is no clear method for stitching these chunks of stereotyped trajectories together [20]. Hence, a drawback of this representation is that it cannot interpret long sequences of movement.

Furthermore, dance scholars have also attacked this problem. Notably, labanotation, created by Rudolf Laban [10], [14], is a method for keeping track of a high degree-of-freedom system. However, it has not integrated with the mainstream dance and is not amenable to robotics as it is nonintuitive and qualitative. Analogously, while many methods can specify robotic control, the framework in this article uniquely aims to bridge this gap and answer this exact dearth of engineering tools for movement analysis and generation and void in our understanding of stylized movement sequences.

In this article, we draw inspiration from the formal principals of movement organization in basic classical ballet. The execution of a few distilled rules produces highly sophisticated and complex motions; hence, this is an excellent candidate movement genre. Using the ability to produce movement sequences as an initial metric for successful representation, we develop machinery to generate coherent movement phrases—in the style of classical ballet.

We extract static poses that are key to the experience of classical ballet and dub them the states of a transition system. Transitions between these states model the movements between poses. This is the subject of [13]. Still, how do we select among these many available pose sequences for one(s) that is both physically feasible and aesthetically meaningful? This problem requires discrete, non physical control methods that can interpret high-level style specifications. Implementing this full framework, we enumerate fundamental rules (hard specifications) and secondary principles (soft specifications) that govern this instantiation of stylized human behavior. In the most basic sense, we are applying system verification and model-checking techniques to a novel system. Our system's unique properties require the full expressiveness afforded by temporal logic statements. But more generally, the framework in this article is sensitive to the void in our understanding of the structure of movement sequences and dearth of tools for movement analysis and generation for robotic systems.

This article extends the work detailed in [13] and [12]. By creating more complex cases studies, we highlight the two-fold goal of this work: On the one hand, we now require the full expressiveness afforded by LTL to implement our stylistic task, and on the other hand, we produce an instantiation of this basic classical ballet movement style.

A Discrete Model

A concept central to ballet's doctrine is that the barre trains and safely warms the muscle groups critical to the correct execution of the freestanding, full-fledged movements that comprise the second portion of class and performances. Hence, these canonical exercises contain the poses and allowable trajectories through them that construct the remaining vocabulary of ballet that is more rich and expressive. The term *barre* refers to the physical hand railing or bar that dancers hold on to balance during the warm-up.

Exercises typically focus on one side of the body, the leg dubbed the working leg, and are repeated twice to work both sides of the body. As such, using the notion of a working leg, we define ten states that correspond to poses in the body's coronal plane (In classical ballet, this restriction still leaves quite a rich vocabulary of movements to describe as many balletic movements make extensive use of this plane.) that are constructed from a triplet of joint angles: the hip, leg, and ankle, as seen in Figure 1. These poses represent shapes critical to the experience of ballet. They are chosen from goal positions at the barre and, as such, are highly recognizable snapshots from the full vocabulary of ballet.

Table 1 lists basic movements from the barre exercises that stitch together series of these goal poses (as demonstrated for développé in Figure 2. The event finishes in an upright standing position after the dancer closes her working leg next to the standing leg.). It follows then that in our model, these movements should correspond to transitions between states, which are also listed in the table. Additionally, we distinguish two transitions for each listed label using a subscript to indicate an in and out variant. The variants stem from the fact that during a movement sequence, a dancer is either on the way out to the goal pose or way back in to a previous one.

Table 1. Basic movements from the barre exercises.

Movement	Transition Label
Plié	plie
Relevé	rele
Battement Tendu	tend
Degajé	dega
Coupé	coup
Frappé	frap
Grand Battement	gran
Passé	pass
Battement	batt
Développé	deve

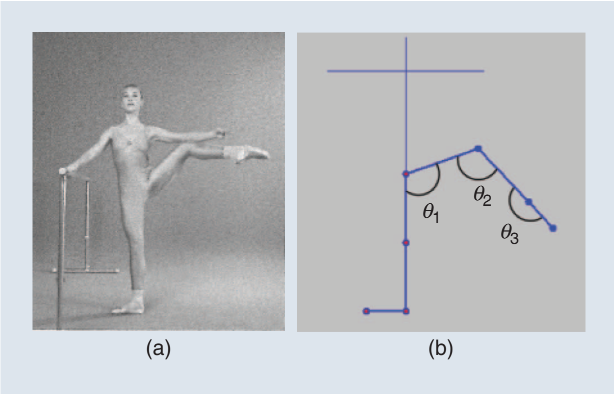


Figure 1. (a) shows a dancer at the barre with her leg in the process of being extended and (b) shows a corresponding abstracted pose. (Photo courtesy of Susan Cook. Reprinted with permission from the University Press of Florida [18].)

Our transition system connects these movements via the appropriate poses in Figure 3. The atomic propositions $\{p_1 \ldots p_9\}$ correspond to poses defined by three joint angles: hip, knee, and ankle. Images of the poses are shown for clarity. For simplicity, we used i to denote state q_i^R and omitted the self-loops at each state. Note that states ten and four satisfy the same propositions, i.e., correspond to the same pose. We differentiate these two related states based on whether the motion of the leg has to remain low (below the hips) or high (at or above hip level) before returning to a neutral state and beginning the next movement. These levels generally correspond to specific movements that stem from this pose as described in [18] and [12]. This system represents allowable movement for one leg only. That is, the resulting language, i.e., the set of all trajectories that start at the initial state, produces feasible barre routines. Some strings might be somewhat unusual (perhaps with movements repeated a strange number of times), but they will certainly be recognizable as being in the style of classical ballet. Formally, we have

$$T_R = (Q_R, q_{0_R}, \rightarrow_R, \Pi_R, h_R), \quad (1)$$

where

1) $Q_R = \{q_1^R, \ldots, q_{10}^R\}$ is the finite set of states;
2) $q_{0_R} = q_2^R$ is the initial state representing the initial pose;
3) $\rightarrow_R \subseteq Q_R \times Q_R$ is the reflexive transition relation (i.e., each state has a self transition);
4) $\Pi_R = \{p_1, \ldots, p_9\} \cup \{\text{Roffground}, \text{Rcoronal}\}$ is a finite set of atomic propositions; and
5) $h_R : Q_R \mapsto 2^{\Pi_R}$ is a satisfaction (output) map, where state q_i^R satisfies the set $h_R(q_i^R)$ of propositions from Π_R as shown in Figure 3.

Correspondingly, we define the transition system that models the motions of the left-leg transition system to be

$$T_L = (Q_L, q_{0_L}, \rightarrow_L, \Pi_L, h_L), \quad (2)$$

where each component is defined as in (1); that is, items 1)–5) are identical for the left-leg transition system with "R" replaced with "L" as necessary. In particular, $\Pi_L = p_1, \ldots, p_9\} \cup \{\text{Loffground}, \text{Lcoronal}\}$.

The atomic propositions are statements that are either true or false about every state of our system. In the next section, we will use the power of temporal logic to observe the evolution of our system in terms of these statements of particular interest; hence, we often think of

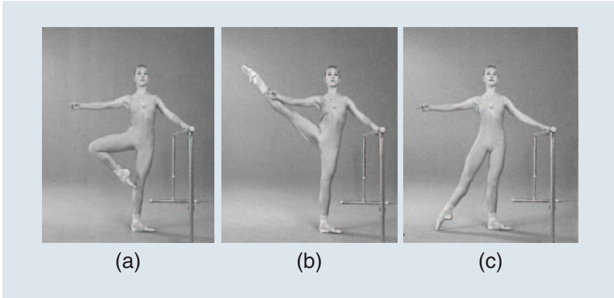


Figure 2. Three snapshots of a développé à la seconde (développé in second position, which best demonstrates the movement in the coronal plane). (a) The pose reached after $coup_o$ and $pass_o$ (state q_5), (b) demonstration of the goal pose (a virtuosic state q_8), which is reached after $deve_o$, and (c) the dancer in the middle of event $batt_i$. (Photo courtesy of Susan Cook. Reprinted with permission from the University Press of Florida [18].)

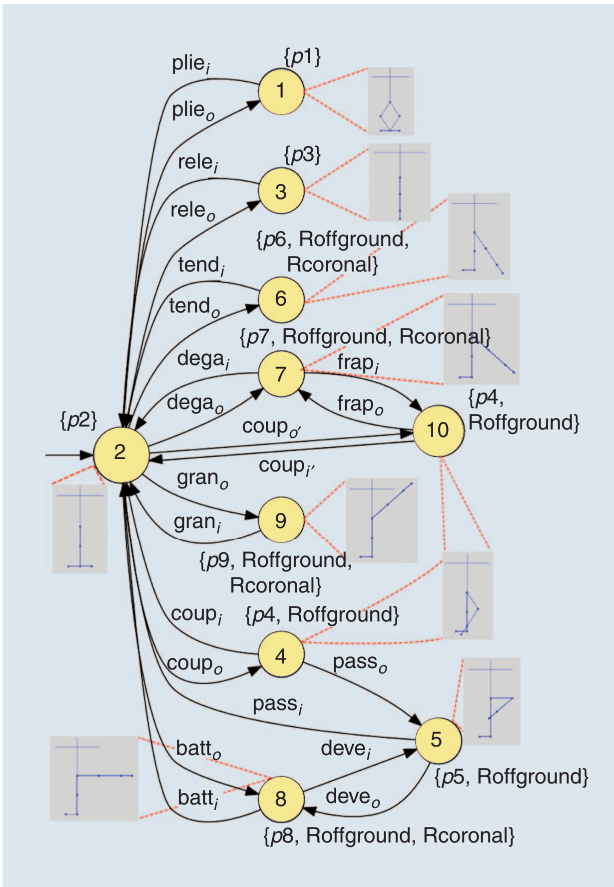


Figure 3. A transition system that models the working leg (right leg) of a dancer during a ballet barre exercise.

these propositions as observations. The pose propositions $\{p_1, \ldots, p_9\}$ for each leg are satisfied only when that leg is in the corresponding pose. State q_{10} is a slight variation of state q_4 and in this two-dimensional framework, we represent them with the same pose. (The pose we call coupe, more formally known as sur le cou-de-pied, has two variations: in one, the foot of the working leg is wrapped around the ankle, while in the other, it is fully pointed and placed next to the ankle. While two states are necessary to prevent nonsense barre sequences from being accepted by the system, the resolution of our model is not intended to capture such subtle differences in pose.) Hence, proposition p_4 is true at states q_4 and q_{10}. The additional propositions, Roffground (the right leg is off the ground) and Rcoronal (the right leg is in a coronal extension away from the body), help script statements about the system more concisely as they apply to several states and have corresponding meaning for the left-leg system. When needed, a script R or L is added to the states and propositions corresponding to T_R and T_L, respectively; however, generally speaking, moves valid (or invalid) on one leg are likewise allowed (or disallowed) on the other so often that these scripts will be neglected as our specifications are symmetrical.

The soft specifications tweak the viable output sequence by enumerating aesthetic guidelines that the system is encouraged to achieve.

Our next task is to compose these two systems using a synchronous product; this composition is liberal and naive because it incorporates every available joint state and transition (some of which are no longer physically possible and/or aesthetically desirable) without taking into consideration whether the composed system is still appropriate. In the next section, we whittle away which of these joint states and transitions the system will inhabit.

More formally, the synchronous product of the two transition systems T_L and T_R, denoted as $T_L \otimes T_R$, is a new transition system with $(Q_P, q_{0_P}, \rightarrow_P, \Pi_P, h_P)$. (Asynchronous transitions of the original single leg systems are also allowed since the reflexive transitions defined in (1) 3) establish self-loops at each state.) The states are Cartesian pairs of the single leg states, i.e., $Q_P \subseteq Q_L \times Q_R$, likewise $q_{0_P} = (q_{0_L}, q_{0_R})$. Transitions exist between these joint states if and only if a transition existed between both single states, i.e., $\rightarrow_P \subseteq Q_P \times Q_P$ is defined by $(q, q') \in \rightarrow_P$ if and only if $q \neq q'$, $(q_L, q'_L) \in \rightarrow_L$, and $(q_R, q'_R) \in \rightarrow_R$, where $q = (q_L, q_R)$ and $q' = (q'_L, q'_R)$. The set of propositions, Π_P, becomes the union of the single leg transitions for the left and right legs with an added proposition, *Spose*, which is satisfied when both legs are in the same positions. The labeling function associates any proposition that was true for either single leg state with the joint state.

Methods for Behavior Specification

Since the one-leg transition system does not contain all the information about the physical capabilities of the robot (T_R assumes that the left leg is static and vice versa), it is entirely possible that the product T_P accepts runs that are not physically possible to execute and not within the range of our goal aesthetic. Hence, we formulate specifications that enable our system to make discrete decisions about viable trajectories (that is, accept or reject various runs through the transition system), where viable is defined in terms of both physical and aesthetic constraints.

To define our problem, we assume that our system is required to satisfy physical constraints of a bipedal geometry and aesthetic requirements of basic classical ballet. Specifically, we want to prevent the robot from executing any physically infeasible runs and are interested in applying aesthetic conditions to the accepted runs of T_P to make them adhere to our chosen dance style. For example, we may disallow a list of two-legged body poses that are perhaps considered ugly as judged by the metric of ballet. We may further influence our output so as to only produce a specific type of movement phrase within the genre that, for example, is typified by more frequent use of certain movements.

Thus, we consider two types of specifications to express the restrictions: 1) hard specifications and 2) soft specifications. A hard specification incorporates physical constraints and aesthetic requirements that the robotic system must satisfy, whereas a soft specification captures certain additional aesthetic requirements that the robotic system is encouraged to achieve. The general philosophy and method for implementing these specifications is provided here; the next section covers our specific choices to generate ballet phrases.

Hard Specifications

Recently, there has been an increasing interest in developing computational frameworks that enable rich specification languages for robotics. In particular, temporal logics, such as linear temporal logic (LTL) and computation tree logic (CTL) have been suggested as motion specification languages [20], [18], [8], [11], [5]. The use of such logics allows for a large spectrum of specifications that include choice of a goal ("go to either A or B"), convergence to a region ("reach A eventually and stay there for all future times"), visiting targets sequentially ("reach A, then B, and then C"), surveillance ("reach A and then B infinitely often"), and the satisfaction of more complicated temporal and logic conditions about the reachability of regions of interest ("Never go to A. Don't go to B unless C or D were visited"). Such robot motion planning and control objectives are achieved based on algorithms inspired from model checking [3] and temporal logic games [17].

Here, we use LTL to express robotic tasks that include an aesthetic component, e.g., "go to goal with grace" or, more specifically, "move in the quick, staccato style of allégro ballet." These tasks have an objective component, such as "take ten steps" and also incorporate subjective qualities like intention and aesthetics, such as "make ten movements that give the impression of being happy." Thus, in this article, through a particular approach to modeling and formal synthesis, we begin to quantify subjective qualities, which are a significant missing link in human-like robot interaction, by scripting them in the established language of LTL.

LTL formulas are built from a set of atomic propositions Π, Boolean operators $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction), and $\rightarrow$ (implication), and temporal operators $\mathbf{X}$ (next), $\mathcal{U}$ (until), $\mathbf{F}$ (eventually), and $\mathbf{G}$ (always). The semantics of LTL formulas are given over infinite words generated by transition systems, such as T_R, T_L, and T_P defined in the "A Discrete Model" section. For simplicity, in what follows, we will denote a general transition system and its set of propositions by T and Π, respectively. A word of T is an infinite sequence over the power set of Π that satisfies the transition relation of T. For example, $\{p_2\}\{p_7, \text{Roffground}, \text{Rcoronal}\}\{p_4, \text{Roffground}\}\{p_2\}\ldots$ is the word generated by a run of T_R that starts at q_2^R, goes to q_7^R, next to q_{10}^R, and then keeps looping among these three states.

A word satisfies an LTL formula ϕ if ϕ is true at the first position of the word; $\mathbf{X}\phi$ states that at the next state, an LTL formula ϕ is true; $\mathbf{F}\phi$ means that ϕ eventually becomes true in the word; $\mathbf{G}\phi$ means that ϕ is true at all positions of the word; and $\phi_1 \mathcal{U} \phi_2$ means ϕ_2 eventually becomes true and ϕ_1 is true until this happens. More expressivity can be achieved by combining the above temporal and Boolean operators (several examples are given later in the article). For example, the word given above satisfies the following LTL formulas: $\mathbf{X}p_7 - p_7$ is true at the next state, $\mathbf{F}p_4 - p_4$ will be true eventually, and $\mathbf{GF}(p_4 \wedge \text{Roffground}) - p_4$ and Roffground are always eventually true.

We use LTL formulas to represent the hard specification. The physical restrictions of the robot and aesthetic requirements of ballet can be translated to LTL formulas as is shown in the "Generating Ballet Phrases" section. To ensure that the hard specifications are achieved in our system output, we augment the product transition system defined in the "A Discrete Model" section with an automata-theoretic representation of the specifications. The resulting structure ensures that a run of our system, which models a movement phrase, obeys both the physical dynamics (represented by T_P) and LTL specification (represented by a formula ϕ). More formally, such a desired movement phrase is a sequence of transitions of T_P that produces a corresponding word satisfying ϕ.

Such a run can be found using techniques inspired by LTL model checking [3], which checks whether all the words of a transition system satisfy an LTL formula ϕ over its set of propositions. Central to LTL model-checking problem is the construction of a Büchi automaton that accepts all and only words satisfying ϕ. An off-the-shelf software tool, such as LTL2BA [7], can be used to generate such a Büchi automaton. The product automaton, $\mathcal{A}$, between the transition system and Büchi automaton accepts all and only the runs of the transition system whose words satisfy ϕ. In other words, this product automaton captures the movement phrases satisfying the hard specification; thus, we call these balletic phrases.

Soft Specifications

The soft specifications tweak the viable output sequence by enumerating aesthetic guidelines that the system is encouraged (instead of forced) to achieve. We define our first type of soft specification, denoted by $\mathcal{S}_1$, as a collection of subsets of Π_P: $\mathcal{S}_1 \subset 2^{\Pi_P}$ [12]. This encodes a set of desired states as follows: we say a state $q \in Q_P$ satisfies $\mathcal{S}_1$ if and only if $h_P(q) \in \mathcal{S}_1$. To broaden this ability to tweak the output, we need to specify desired transitions. Accordingly, we define a new, second type of soft specification as:

$$\mathcal{S}_2 = \{(prop_i, prop\prime_i)\}, \text{where } prop_i, prop\prime_i \in 2^{\Pi_P}. \quad (3)$$

Again, we say a transition $(q, q\prime) \in \rightarrow_P$ satisfies $\mathcal{S}_2$ if and only if $(h_P(q), (h_P(q\prime)) \in \mathcal{S}_2$. Hence, the collections $\mathcal{S}_1$ and $\mathcal{S}_2$ model a desire for the robotic system to prefer certain states (poses) and transitions (movements) more than others. The technique that allows these specifications to guide our output is outlined next; the faculty this adds to our framework is demonstrated in the "Generating Ballet Phrases" section.

Implementation

We design a (local, real time) receding-horizon controller to find a run that maximizes rewards collected based on local information obtained at the current state while guaranteeing the satisfaction of the hard specification. Using an approach similar to that in [5], where a receding-horizon controller was designed, we employ a measure of progress toward satisfying the LTL formula ϕ. If the controller is designed to always increase this progress, as defined by our measure, then we can show that ϕ is satisfied. The proposed approach relied on an assignment of 1) a suitable cost to each state of $\mathcal{A}$ (to ensure that the output sequence adheres to ϕ as the sequence is selected) and 2) rewards on states and transitions that satisfy the propositions in S_1 and S_2 (to influence the poses and motions in the output sequence).

As previously discussed, the product automaton $\mathcal{A}$ is constructed from the transition system T_P and Büchi automaton corresponding to ϕ; this object jointly encodes the system dynamics and hard specifications. It is defined as a tuple $(S_{\mathcal{A}}, S_{\mathcal{A}0}, \delta_{\mathcal{A}}, \omega_{\mathcal{A}}, F_{\mathcal{A}})$, where $S_{\mathcal{A}}$ is the set of states, $S_{\mathcal{A}0}$ the set of initial states, $\delta_{\mathcal{A}} : S_{\mathcal{A}} \mapsto 2^{S_{\mathcal{A}}}$ the transition function, $\omega_{\mathcal{A}} : S_{\mathcal{A}} \times S_{\mathcal{A}} \mapsto R^+$ a positive-valued weight function inherited from T_P, and $F_{\mathcal{A}}$ the set of accepting states. We denote by $d(p_i, p_j)$ the shortest distance (based on the weight of the transitions) from state $p_i \in S_{\mathcal{A}}$ to $p_j \in S_{\mathcal{A}}$. The shortest distance between two states can be efficiently computed by a shortest path algorithm such as Dijkstra's algorithm [15].

The cost assignment (Note that the costs associated with each state are computed offline and then saved in a look-up table to be used later in conjunction with the real-time controller.) for all states $p_i \in S_{\mathcal{A}}$ is defined as

$$J(p_i) = \begin{cases} \min_{p_j \in F_{\mathcal{A}^*}} d(p_i, p_j), & \text{if } p_i \notin F_{\mathcal{A}^*} \\ 0, & \text{if } p_i \in F_{\mathcal{A}^*} \end{cases}, \quad (4)$$

where $F_{\mathcal{A}^*}$ is the largest self-reachable subset of $F_{\mathcal{A}}$. This cost encodes the minimum distance from states in $\mathcal{A}$ to the set $F_{\mathcal{A}^*}$. We use this cost to keep our controller returning to these states since acceptance of a run through $\mathcal{A}$ requires that the run intersect this set infinitely often.

We introduce rewards (positive real numbers) to the states and transitions indicated by $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. When the collections $\mathcal{S}_1$ and $\mathcal{S}_2$ are not empty, the states and transitions satisfying these sets have the potential to produce rewards. The rewards are generated in a time-varying fashion, and the appearance and disappearance of rewards and their values are randomized. Upon visiting a state or taking a transition, the corresponding reward is collected and subsequently consumed. By maximizing the collected rewards, we encourage the system to visit desired states and take preferred transitions more frequently. We make the natural assumption that, at each time instant, the system can foresee only a few steps ahead (analogous to a dancer who is performing a free, improvised solo and planning just a few steps ahead at a time). Thus, the run will contain infinite repetitions of finite runs that maximize the total reward collected over the finite runs with path length not more than a planning horizon length L.

A hard specification incorporates physical constraints and aesthetic requirements that the robotic system must satisfy.

The controller is a combination of a precontroller and a postcontroller. The precontroller drives the system from any state $p \notin F_{\mathcal{A}^*}$ to a state p', where $J(p') < J(p)$, while maximizing the rewards collected locally. The repeated executions of the precontroller will drive the system from a state $p \notin F_{\mathcal{A}^*}$ to a state in $F_{\mathcal{A}^*}$ in a finite number of transitions. The postcontroller drives the system from a state in $F_{\mathcal{A}^*}$ to a state outside $F_{\mathcal{A}^*}$, while also maximizing the local rewards. The controller switches between the precontroller and postcontrollers (depending on the current state) infinitely many times so that the accepting states of $\mathcal{A}$ is visited infinitely many times, and thus the produced infinite run is accepted by $\mathcal{A}$. Since $\mathcal{A}$ accepts all and only the runs of T_P whose words satisfy ϕ, the run produced by the proposed controller can be used to obtain a run on T_P that favors the desired states and transitions and generates a word satisfying ϕ (i.e., a desired movement phrase).

Generating Ballet Phrases

The hard specifications, which are applied to both case studies and ensure that the basic style objectives are met, are given according to a collection of physically and aesthetically driven rules (which are presented here with minor adjustments from those in [12]) converted to LTL formulas as follows:

1) We first ensure that no physically infeasible joint poses are entered by the system. Such poses include those that i) imply impossible jumping positions, i.e., state (q_6, q_7), ii) are impossible to hold because they are neither jumping poses nor feasible standing poses, i.e., both legs attempting to touch the ankle of the other as in state (q_4, q_4), and iii) the geometry of the biped's hips will not easily allow and thus form awkward poses that are ugly in the eyes of the traditional ballet choreography, i.e., state (q_1, q_3). Thus, sets of poses are disallowed:
   i) Always avoid both legs off ground except poses in which two legs are in the same position, and poses satisfying $p_7 \wedge p_{10}$, $p_5 \wedge p_8$, or $p_4 \wedge p_5$.
   $$\mathbf{G}\ \neg(\text{Roffground} \wedge \text{Loffground} \wedge (\neg\text{Spose}) \wedge \neg(p_7 \wedge p_{10}) \wedge \neg(p_5 \wedge p_8)) \wedge \neg(p_4 \wedge p_5)).$$
   ii) Always avoid both legs in poses p_4, p_6, and p_{10}.
   $$\mathbf{G}\ \neg(\text{Spose} \wedge (p_4 \vee p_6 \vee p_{10})).$$
   iii) Always avoid both legs in poses satisfying $p_1 \wedge p_2$, $p_1 \wedge p_3$, or $p_2 \wedge p_3$.
   $$\mathbf{G}\ \neg((p_1 \wedge p_2) \vee (p_1 \wedge p_3) \vee (p_2 \wedge p_3)).$$
2) Next, we prevent the system from switching to a pose which, given a current pose, would cause a biped to fall over. We restrict transitions from states that have only one leg supporting the body; these include i) jumping from a leg in relevé—this would require just the toe joint of one leg to lift the entire body weight into the air, i.e., (q_3, q_8) to (q_8, q_5), ii) extending the supporting leg when it is in plié, i.e., (q_1, q_5) to (q_8, q_5), and iii) extending the supporting leg when it is on flat, i.e., (q_2, q_5) to

(q_8, q_5). Thus, a collection of sequences of poses are disallowed:

i) When one leg is in pose p_1 and the other leg is already off ground, always avoid lifting both legs off ground.

$$\mathbf{G}\ (\neg((p_1 \wedge (\text{Roffground} \vee \text{Loffground})) \rightarrow \mathbf{X}\ (\text{Roffground} \wedge \text{Loffground}))).$$

ii) When the right leg is in pose p_2 and left leg is currently off ground, always avoid performing coronal extensions using the right leg without putting down the left leg first.

$$\mathbf{G}((p_2 \wedge \text{Roffground}) \rightarrow \mathbf{X}\neg(\text{Roffground} \wedge \text{Lcoronal})).$$

iii) When the left leg is in pose p_2 and right leg is currently off ground, always avoid performing coronal extensions using the left leg without putting down the right leg first.

$$\mathbf{G}((p_2 \wedge \text{Loffground}) \rightarrow \mathbf{X}\neg(\text{Loffground} \wedge \text{Rcoronal}).$$

3) Finally, we ensure that the system puts both legs standing on the ground once in a while. This is phrased as a requirement to visit the standing pose, state (q_2, q_2), infinitely often: "visit the pose where both legs are in pose p_2 infinitely many times."

$$\mathbf{G\,F}\ (\text{Spose} \wedge p_2).$$

Now, we will use our soft specifications and an optimal controller to produce sequences that satisfy more subtle style objectives as explained in the "Methods for Behavior Specification" section.

We first provide a case study that demonstrates how small changes in style specifications can lead to a sequence with a new emotive effect. Ultimately, it is this type of coupling between choreographic structure and emotive effect that this framework aims to crystallize. This case study demonstrates the effect of changing just one aspect of a specification. We encourage the movement called développé shown in Figure 2. This movement does not specify the configuration of the other leg (the supporting leg). In fact, the supporting leg may either be in pose 1 (plié), 2 (flat), or 3 (relevé)—a choice left up to the choreographer.

The choice of the standing leg position determines whether a sharp depression or elevation of the dancer's body occurs. Changing levels is a well-known tool in choreography: if an entire phrase of movement occurs on the same level, it may be perceived as boring or monotone. Adding elevation can be thought of as adding high notes in a musical score that typically brings images that are light and happy to the audience's mind. On the other hand, low notes on the bass clef in music imbue a more somber, mysterious tone to the score.

The ability of the machinery presented in this article to produce such sequences is demonstrated in Figure 4 and further analyzed in Figure 5. In these cases, we have added

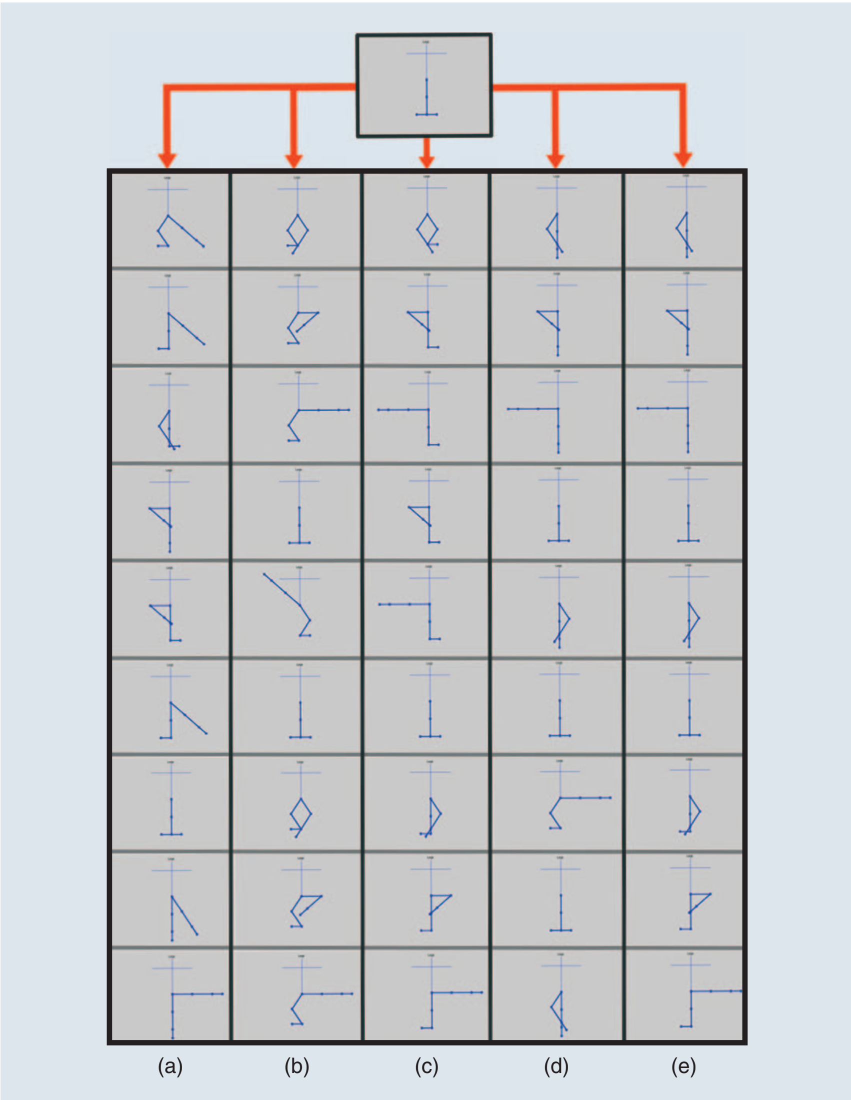


Figure 4. Five sequences demonstrate how a simple change in specification produces great variation in the output sequence. (a) Case 0 (base case), where only the hard specifications are applied; (b) Case 1 encourages more développés in plié; (c) Case 2 encourages développé on flat; (d) Case 3, with elevated développés on relevé; and (e) Case 4, where the développé is encouraged disregarding the standing leg's position.

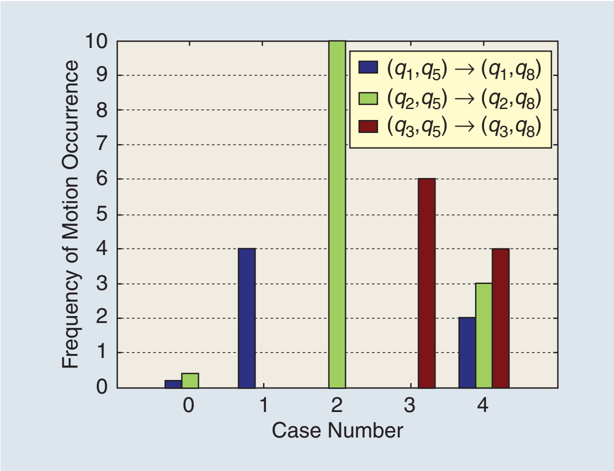


Figure 5. Case 0 shows the average of ten runs without any soft specification, showing the tendency of the system without any encouragement from soft specifications. The occurrences of developpe in plie, on flat, and on releve are shown for the remaining four cases in blue, green, and red, respectively. Each sample path contains 52 steps.

rewards directly to our composed transition system. These soft specifications encourage specific movements to occur and are not phrased in LTL statements like the hard specifications—instead, we employ our receding-horizon controller to collect the rewards on transitions and thus produce a desired output sequence. Namely, in each case we encourage the transitions coup_o, pass_o, and deve_o when the standing leg is in the desired pose. For Case 1, this corresponds to six transitions: $(q_2, q_2) \to (q_1, q_4)$, $(q_1, q_4) \to (q_1, q_5)$, $(q_1, q_5) \to (q_1, q_8)$, and the three transitions that correspond to performing the movement on the other leg. The soft specification can be summarized as in (3).

Second, we present a case study where, by modifying the hard specifications, we encourage full sequences of desired movements rather than single poses as in [12] or transitions as presented above. Specifically, we add another rule that translates to an LTL formula as given as follows and encourage the pose (q_7, q_7):

"Any time a glissade happens, follow it with plié grand jeté or plié pas de chat développé."

$$\mathbf{G}((p_7 \wedge \text{Spose}) \to \mathbf{X}((p_2 \wedge \text{Spose}) \wedge \mathbf{X}((p_1 \wedge \text{Spose}) \wedge \mathbf{X}((p_2 \wedge \text{Spose}) \wedge \mathbf{X}(((p_2 \wedge p_4) \wedge \mathbf{X}((p_4 \wedge p_5) \wedge \mathbf{X}(p_5 \wedge p_8))) \vee (p_8 \wedge \text{Spose})))))).$$

The encouraged sequences of movement, glissade plié grand jeté, corresponds to $(q_7, q_7)(q_2, q_2)(q_1, q_1)(q_2, q_2)(q_8, q_8)$ and the other, glissade plié pas de chat développé, corresponds to $(q_7, q_7)(q_2, q_2)(q_1, q_1)(q_2, q_2)(q_2, q_4)(q_4, q_5)(q_5, q_8)(q_8, q_8)$. Note that the specification is symmetric; hence, a sequence with the left and right leg roles reversed is also encouraged.

With this added hard specification, we are able to ensure that one of these movement sequences will occur every time both legs are in pose 7. Encouraging this pose in the soft specification will thus encourage the frequency of the desired sequences. The results from this case study are presented in Figure 6. The resulting sequences exhibit frequent use of this series of jumps that may comprise the virtuousic displays that characterize certain sections of evening length ballets.

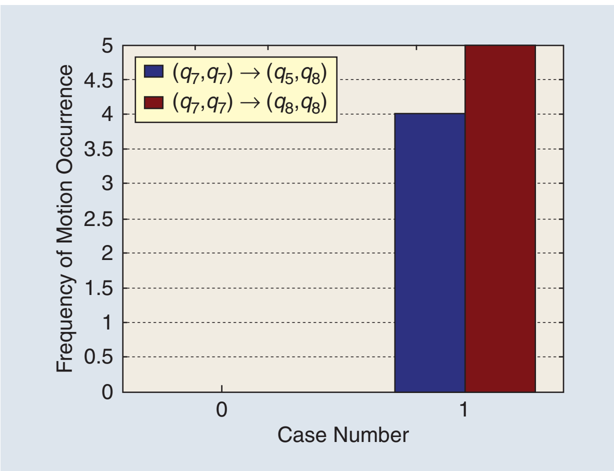


Figure 6. Occurrences of the transitions satisfying the soft specification. Each output sequence contained 100 steps. Without the specification, neither sequence showed up in the output sequence. With the specification, glissade plié pas de chat développé (blue) appeared four times and glissade plié grand jeté (red) appeared five times.

Conclusions and Outlook

In this article, we have specified a grammar for leg positions in ballet movements restricted to the coronal plane. Furthermore, our specification is guaranteed to follow certain principles, both physical and aesthetic, through the expressive power of LTL. This primary contribution may lead to systems (humanoid or otherwise) that behave in a way that is natural for its given context. A deterministic program can limit the flexibility of a system's ability to cope and adapt to its environment. When implemented on a robotic system, a behavior, as outlined here, provides one or more states (perhaps a preferred state) that the system may enter given its current state.

A novel venue for movement and style analysis is the second contribution of this model. The movement analysis required to produce a system capable of automatically generating movement phrases in the style of classical ballet results in a quantitative phrasing of the rules that govern this somewhat curious example of human behavior. A quantitative survey of specific dance styles (between genres, i.e., flamenco versus hip hop, and choreographers i.e., Mark Morris versus Twyla Tharp) would bolster and perhaps corroborate years of qualitative dance study that hold that specific movement patterns evoke different aesthetic and emotive effects in dance choreography.

From a dancer's perspective, the lack of tools for representing human movement induces quite an imbalance: a musician works with notes arranged in octaves, chords, and scales; an artist paints with a palette of colors known to span the space of light that is perceivable to humans; and writers knit nouns and verbs into clauses, sentences, and paragraphs. Movements are certainly classified and named (i.e., pirouette and brisé volé), and some words from these other artistic disciplines have filtered into the dance world to describe conglomerate structures of motion (phrases and movements). In this sense, there is qualitative organization in artistic movement, but can we as engineers develop analytical tools to justify or edify any of this them?

The extension of this philosophy into a system with well-defined inputs and outputs presents interesting questions for systems theory, and as has been presented in this article, systems theory allows a new articulation of the creative process involved in choreographing human movement. We see the success of this model as twofold: we phrase the 1) states of our system in terms of natural body poses and 2) evolution of these states in terms of the most meaningful facts about the system via LTL.

In summary, ballet is a highly ordered behavior of a truly complex biological system whose attributes have important analogs in systems theory that warrant quantitative study. By formulating aesthetic style from a systems theoretic perspective and, thus, resolving the attributes of human movement that typify and comprise stylized movement, we are beginning to define a metric for a previously abstract concept. Furthermore, the structure of the aesthetic movement explored here provides an interesting challenge for robotics research and formal methods, namely that of how the composition of structured discrete event systems may generate desired behavior for humanoid robotic tasks.

Acknowledgment

The work by LaViers and Egerstedt is supported by the U.S. National Science Foundation through grant number 0757317. The work by Chen and Belta is supported by ONR MURI N00014-09-1051, ARO W911NF-09-1-0088, AFOSR YIP FA9550-09-1-020, and NSF CNS-0834260.

References

[1] E. Bradley and J. Stuart, "Using chaos to generate variations on movement sequences," *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 8, no. 4, pp. 800–807, 1998.

[2] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Proc. 1997 IEEE Computer Society Conf. Computer Vision and Pattern Recognition.* New York: IEEE, 2002, pp. 568–574.

[3] E. M. M. Clarke, D. Peled, and O. Grumberg, *Model Checking*. Cambridge, MA: MIT Press, 1999.

[4] D. Del Vecchio, R. M. Murray, and P. Perona, "Decomposition of human motion into dynamics-based primitives with application to drawing tasks* 1," *Automatica*, vol. 39, no. 12, pp. 2085–2098, 2003.

[5] X. C. Ding, C. Belta, and C. G. Cassandras, "Receding horizon surveillance with temporal logic specifications," in *Proc. 49th IEEE Conf. Decision and Control*, 2010, pp. 256–261.

[6] C. Fanti, L. Zelnik-Manor, and P. Perona, "Hybrid models for human motion recognition," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. New York: IEEE, 2005, pp. 1166–1173.

[7] P. Gastin and D. Oddoux, "Fast LTL to Buchi automata translation," in *Proc. 13th Int. Conf. Computer Aided Verification*, LNCS, G. Berry, H. Comon, and A. Finkel, Eds. Berlin: Springer-Verlag, 2001, pp. 53–65.

[8] H. K. Gazit, G. Fainekos, and G. J. Pappas, "Where's Waldo? Sensor-based temporal logic motion planning" in *Proc. IEEE Conf. Robotics and Automation*, Rome, Italy, 2007, pp. 3116–3121.

[9] M. J. Gielniak, C. K. Liu, and A. L. Thomaz, "Stylized motion generalization through adaptation of velocity profiles," in *Proc. IEEE RO-MAN.* New York: IEEE, 2010, pp. 304–309.

[10] A. Hutchinson and D. Anderson, *Labanotation*. New York: New Directions, 1954.

[11] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. Automat. Contr.*, vol. 53, no. 1, pp. 287–297, 2008.

[12] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, "Automatic generation of balletic motions," in *Proc. ACM/IEEE Second Int. Conf. Cyber-Physical Systems*, 2011, pp. 13–21.

[13] A. LaViers and M. Egerstedt, "The ballet automaton: A formal model for human motion," in *Proc. American Control Conf.*, 2011, pp. 3837–3842.

[14] V. Maletic, *Body Space Expression*. Berlin: Walter de Gruyter & Co., 1987.

[15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization Algorithms and Complexity*. New York: Dover, 1998.

[16] B. Peng, G. Qian, and S. Rajko, "View-invariant full-body gesture recognition from video," in *Proc. 19th Int. Conf. Pattern Recognition (ICPR'08).* New York: IEEE, 2009, pp. 1–5.

[17] N. Piterman, A. Pnueli, and Y. Saar, "Synthesis of reactive(1) designs," in *Proc. VMCAI*, Charleston, SC, 2006, pp. 364–380.

[18] G. W. Warren and S. Cook, *Classical Ballet Technique*. Gainesville, FL: Univ. South Florida Press, 1989.

[19] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Proc. Int. Conf. Hybrid Systems: Computation and Control*, 2010, pp. 101–110.

[20] S. Gentry and E. Feron, "Modelling musically meaningful choreography," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 2004, vol. 4, pp. 3880–3885.

Amy LaViers, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. E-mail: alaviers@gatech.edu.

Yushan Chen, Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215, USA. E-mail: yushanc@bu.edu.

Calin Belta, Department of Mechanical Engineering, Division of Systems Engineering, and the Bioinformatics Graduate Program, Boston University, MA 02215, USA. E-mail: cbelta@bu.edu.

Magnus Egerstedt, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. E-mail: magnus@gatech.edu.