UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Coordination Dynamics in Human-Robot Teams**

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Tariq Iqbal

Committee in charge:

Professor Laurel D. Riek, Chair
Professor Manmohan Chandraker
Professor Andrea Chiba
Professor Henrik Christensen
Professor Ryan Kastner

2017

The dissertation of Tariq Iqbal is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2017

DEDICATION

To my parents, Delwara Begum and Md. Jalal Uddin,

and my brother, Arif Iqbal

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to take the opportunity to thank several individuals who extended their direct and indirect influence and support during my Ph.D. process.

I was fortunate to have Prof. Laurel Riek, one of the nicest persons I have ever met, as my advisor. She helped me to understand the true meaning of research. She always inspired me to think out of the box and gave me the freedom to explore new research ideas, even when they were not entirely related to my core area of research. I feel incredibly lucky to be a member of her group. I also want to thank my dissertation committee members, Prof. Manmohan Chandraker, Prof. Andrea Chiba, Prof. Henrik Christensen, and Prof, Ryan Kastner for their valuable comments and suggestions toward my dissertation.

I want to thank my labmates, Michael Gonzales, Cory Hayes, Darren Chan, Angelique Taylor, Maria O'Connor, and Maryam Moosaei for their continuous help and motivation during the ups and downs in graduate school. I also want to thank Samantha Rack for her help during the development process of the anticipation algorithms.

I am very thankful to my friends, S. M. Iftekharul Alam, Afzal Hossain, Atif Hasan Rahman, Swakkhar Shatabda, and Md. Moinul Islam for their continuous support. I cannot think of any decision that I have taken in last decade without first consulting with them. Without maintaining constant communication with them, it would have been hard for me to stay focused.

I would like to take the opportunity to thank my parents, Delwara Begum and Md. Jalal Uddin, and my brothers, Ashif Iqubal and Arif Iqbal. They always provided me with opportunities for the best possible education, and have always been there to support me.

Last but most importantly, I want to thank my wife, Tahmina Afrin Liza, for her unconditional sacrifice and support since we first met. I am thankful to her for coming to

a foreign country with me, where she is not even allowed to work, despite her having an advanced degree and has the qualifications to work anywhere she pleases. I am also thankful to her for bringing the best possible inspiration in our life, Aarjo. It would not be possible for me to stay motivated without them on my side throughout my Ph.D. process.

My research was made possible by funding support from the National Science Foundation. Chapter 3 of this dissertation contains material from "A Model for Time-Synchronized Sensing and Motion to Support Human-Robot Fluency," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Workshop on Timing in HRI, 2014 [74]. The dissertation author was the primary investigator and author of this paper.

Chapters 4 and 5 contain material from "Assessing group synchrony during a rhythmic social activity: A systemic approach," by T. Iqbal and L. D. Riek, which appears in Proceedings of the Conference of the International Society for Gesture Studies (ISGS), 2014 [80]. The dissertation author was the primary investigator and author of this paper.

Chapters 4 and 5 contain material from "A Method for Automatic Detection of Psychomotor Entrainment," by T. Iqbal and L. D. Riek, which appears in IEEE Transaction on Affective Computing, 2015 [82]. The dissertation author was the primary investigator and author of this paper.

Chapters 5 and 6 contain material from "Detecting and Synthesizing Synchronous Joint Action in Human-Robot Teams," by T. Iqbal and L. D. Riek, which appears in Proceedings of the International Conference on Multimodal Interaction, 2015 [83]. The dissertation author was the primary investigator and author of this paper.

Chapter 6 contains material from "Role distribution in synchronous human-robot joint action," by T. Iqbal and L. D. Riek, which appears in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, Towards a Framework for Joint Action Workshop, 2014 [81]. The dissertation author was the

primary investigator and author of this paper.

Chapter 6 contains material from "Mobile robots and marching humans: Measuring synchronous joint action while in motion," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the AAAI Fall Symposium on AI-HRI, 2014 [75]. The dissertation author was the primary investigator and author of this paper.

Chapter 6 contains material from "Joint action perception to enable fluent human-robot teamwork," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 2015 [76]. The dissertation author was the primary investigator and author of this paper.

Chapters 7 and 8 contain material from "Will you join the dance? Toward synchronous joint action in human robot teams," by T. Iqbal., S. Rack, and L. D. Riek, which appears in Proceedings of the Conference of Joint Action Meeting, 2015 [78]. The dissertation author was the primary investigator and author of this paper.

Chapters 7 and 8 contain material from "Movement coordination in human-robot teams: A dynamical systems approach," by T. Iqbal., S. Rack, and L. D. Riek, which appears in IEEE Transactions on Robotics, 2016 [79]. The dissertation author was the primary investigator and author of this paper.

Chapter 9 contain material from "Human coordination dynamics with heterogeneous robots in a team," by T. Iqbal. and L. D. Riek, which appears in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Pioneers workshop, 2016 [84]. The dissertation author was the primary investigator and author of this paper.

Chapter 9 contains material from "Coordination dynamics in multi-human multi-robot teams," by T. Iqbal. and L. D. Riek, which appears in IEEE Robotics and Automation Letters, 2017 [87]. The dissertation author was the primary investigator and author of this paper.

Chapter 10 contains material from "Tempo Adaptation and Anticipation Methods for Human-Robot Teams," by T. Iqbal., M. Moosaei, and L. D. Riek, which appears in Proceedings of the Robotics: Science and Systems Conference, Planning for HRI: Shared Autonomy and Collab. Robotics Work, 2016 [77]. The dissertation author was the primary investigator and author of this paper.

Chapter 10 contains material from "Can robots synchronize with humans in tempo changing environments?" by T. Iqbal. and L. D. Riek, which appears in Proceedings of the Conference of Joint Action Meeting, 2017 [86]. The dissertation author was the primary investigator and author of this paper.

# VITA

| | |
|---|---|
| 2007 | B. S. in Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh |
| 2012 | M. S. in Computer Science, University of Texas at El Paso, Texas |
| 2017 | Ph. D. in Computer Science, University of California San Diego, California |

ABSTRACT OF THE DISSERTATION

**Coordination Dynamics in Human-Robot Teams**

by

Tariq Iqbal

Doctor of Philosophy in Computer Science

University of California, San Diego, 2017

Professor Laurel D. Riek, Chair

As robots become more common in our daily lives, they will be expected to interact with and work with teams of people. If a robot has an understanding of the underlying dynamics of a team, then it can recognize, anticipate, and adapt to human motion to be a more effective teammate.

To enable robots to understand team dynamics, I developed a new, non-linear method to detect group synchronization, which takes multiple types of discrete, task-level events into consideration. I explored this method within the context of coordinated action and validated it by applying it to both human-only and mixed human-robot teams. The results suggest that our method is more accurate in estimating group synchronization

than other methods from the literature.

Building on this work, I designed a new method for robots to perceive human group behavior in real-time, anticipate future actions, and synthesize their motion accordingly. I validated this approach within a human-robot interaction scenario, where a robot successfully and contingently coordinated with people in real-time. We found that robots perform better when they have an understanding of team dynamics than they do not.

Moreover, I investigated how the presence and behavior of robots affect group coordination in multi-human, multi-robot teams. The results suggested that group coordination was significantly degraded when a robot joined a human-only group, and was further degraded when a second robot joined the team and employed a different anticipation algorithm from the other robot. These findings suggest that heterogeneous behavior of robots in a multi-human group can play a major role in how group coordination dynamics change.

Furthermore, I designed and implemented algorithms for robots to coordinate with people in tempo-changing environments. These algorithms leveraged a human-like understanding of temporal anticipation and adaptation during the coordination process. I validated the algorithms by applying them in a human-robot drumming scenario. The results suggest that an adaptation process alone enables a robot to achieve human-level performance. Moreover, by combining anticipatory knowledge (anticipation algorithm), along with an adaptation process, a robot can be even better than people in both uniform and single tempo-changing conditions.

My research will enable robots to recognize, anticipate, and adapt to human groups. This work will help enable others in the robotics community to build more fluent and adaptable robots in the future, and provide a necessary understanding for how we design future human-robot teams.

# Chapter 1

# Introduction

As technology advances, autonomous robots are becoming more involved in human society in a variety of roles. Robotic systems have long been involved in assembly lines automating and increasing efficiency of dexterous factory procedures [223]. Their involvement is becoming prominent in other situations, like assistive living, surgery, physical therapy etc [44, 123, 20]. In recent years, *assistive robots* are becoming more common in healthcare situations, like robotic wheelchairs to aid people with physical mobility [52], robotic exoskeletons to help body control [97], and rehabilitation robots to provide physical therapy after stroke [112].

However, as robots leave controlled spaces and begin to work alongside people, many things taken for granted in robotics concerning perception and action do not apply, as people act unpredictably, e.g., they "break the rules" when it comes to what a robot can expect *a priori*. In order for robots to effectively perform their tasks and integrate into Human Social Environments (HSEs), they must be able to comprehend high-level social signals of the group and respond appropriately [176]. While recent advances in the field have improved robot perception in general (c.f., [17]), it is still difficult for a robot to recognize each human action in the real world, and use that information in a timely

fashion to make informed decisions about their own actions [229].

As robots are becoming prominent in society, they are expected to interact with humans in teams. For example, a robot might encounter people in a group performing various social actions, such as engaging in social games or synchronized movements. It can be difficult for a robot to perceive and understand all of the different movements of the humans to make effective decisions as a team mate. If a robot could make better sense of how humans interact among themselves in a group, its interactions with humans would reach a higher level of coordination, resulting in a *fluent* meshing of actions [180, 74, 75, 64, 18].

Group interaction is an important aspect of human social interaction, and is an active area of research [19, 91, 14, 10, 49, 47]. In group interaction, the activities of each member continually influence the activities of other group members. Most groups create a state of interdependence, where each member's outcomes and actions are determined in part by other members of the group [47]. This process of influence can result in coordinated group activity over time, which can be described as the synchronization of the group.

In many group interaction scenarios, humans coordinate their activities with the rest of the group members, and the group eventually reaches to a synchronous state. Synchronous activity, or synchronous joint action, can be described as the spatiotemporal coordination of signals, which results from rhythmic responsiveness to a perceived rhythmic signal [159]. This is a form of social interaction where two or more participants coordinate their actions in space and time while making changes to their environment [201]. Synchronous joint action is an important behavioral indicator of group-level cohesiveness, and also important for accurately understanding the affective behavior of a group [172, 218]. It is also a key component of human audio-visual attention and direction, and of social learning and cognitive development [185, 163].

Synchronous joint action can occur in many group contexts, such as dance, music, and games, and can be a result of both intentional and unintentional motor coupling during social interaction. *Intentional entrainment* occurs in cooperative group tasks, such as rowing a boat with a team. In contrast, an example of *unintentional entrainment* would be a group of friends spontaneously coordinating their movements when walking together [217, 172]. In general, humans are skilled at coordinating their movements in group situations.

In a group setting, the synchronization phenomenon can be observed not only based on the coordinated movements of the group members, but also on how the group members perform activities. If a group task is cooperative and rhythmic (e.g., synchronous swimming), then synchronization can also be found in the activity performed by each member of the group. Thus, systemic approaches are proposed in the literature to model the synchronization of a group, both at the individual-level and group-level.

Understanding human team dynamics is not trivial. One of the main challenges is the unpredictability in human behavior, little of which can be known in advance. Another major challenge is accurately detecting human activities "in the wild" [209], e.g., in ecologically realistic settings.

However, if a robot has some ability to model team dynamics, then it can anticipate future actions in a team, and adapt to those actions in order to be an effective teammate. Synchronous joint action is a kind of a high-level behavior that robots will encounter in a group while teaming with humans. Recognizing synchronous joint action and adapting to it will help the robots be more *fluent* during interaction, and thus be more acceptable by human counterparts.

## 1.1 Motivation and scope

Researchers in the fields of psychology, cognitive science, kinesiology, dance and music have explored how humans interact among themselves within a group [150, 198, 102, 125, 12, 115, 172]. However, there is a lack of an automatic tool to model and understand coordination from the task-level events of the group members. Having such an automatic tool will be beneficial for all these fields, as well as it will push the science forward.

In robotics, understanding human group interaction and joint action has recently surged in interest in the field of human-robot interaction (HRI) [116, 138, 149, 133, 162, 58, 67, 42, 137, 94, 139]. Understanding group dynamics will be helpful for robots to synthesize their own movements in teams.

Typically, human-robot teaming research has been more geared toward dyadic HRI (one human, one robot), and has focused more on manipulation tasks. In contrast, the goal of this work is to enable robots to interact fluidly within groups of people by understanding team dynamics from the gross action rather than dexterous manipulation [74, 75]. This is an important gap to address, as a large amount of human activity takes place in groups, and within groups there is a higher likelihood of synchronous activity occurring [172]. Thus, robots that can be aware of such activity and determine how best to engage are more likely to work successfully and contingently with humans, as well as be more accepted by humans [176].

There are many dimensions to this problem, and this dissertation explores the following aspects of them:

1. How to measure the degree of synchronization in a group

2. How to validate this measure, both in the context of human-human, and human-robot team scenarios

3. How a robot can use this understanding to anticipate future activities of the group members and inform its own movements accordingly

4. How a robot can anticipate and adapt to a tempo-changing rhythm during a synchronous group activity

5. What the effects are of varying the number of robots with different behaviors in multi-human multi-robot teams

6. How robots can leverage a human-like understanding of temporal anticipation and adaptation when coordinating with people

## 1.2 Contributions

The contributions of this work are as follows:

- **Developed a new non-linear method to measure the synchronization of a group.** Our method takes multiple types of discrete, task-level events into consideration, to automatically detect the degree of synchronization of a group. It is capable of modeling synchronous and asynchronous behavior of the group, and is well-matched to the group members' perception about synchrony. Our method is more accurate in estimating synchrony than other methods from the literature that depend on a single event type [82].

- **Validated the new method by applying it to a human-human team.** We validated the method by applying it to a human-human team, where the group played a multiple event-based rhythmic game. During this game each player performed a periodic psychomotor activity within the group, and each player's movements influenced all others' movements. The results suggest that the method successfully

measured the degree of synchronization of the group in 100% of cases, which matches with the collective perception of the group members [82].

- **Validated the new method by applying to a human-robot team.** We validated our method using it to study people engaging in synchronous and asynchronous marching tasks while being followed by autonomous mobile robots. We again found that our method is accurate in measuring group synchrony [76].

- **Designed a new approach for robots to anticipate and synthesize coordinated actions.** We have designed a new approach to enable robots to perceive human group motion in real-time, anticipate future actions, and synthesize their own motion accordingly. We explore this within the context of joint action, where humans and robots move together synchronously. We validate the method within a human-robot interaction scenario, where an autonomous mobile robot observes a team of human dancers, and then successfully and contingently coordinates its movements to "join the dance". We compared the results of our anticipation method to move the robot with another method which did not rely on team dynamics, and found our method performed better both in terms of more closely synchronizing the robot's motion to the team, and also exhibiting more contingent and fluent motion. These findings suggest that the robot performs better when it has an understanding of team dynamics than when it does not [79].

- **Investigated how the presence and behavior of robots affect group coordination in multi-human multi-robot teams.** I have investigated how the presence of robots affect group coordination when both the anticipation algorithms they employ and their number (single robot or multiple robots) vary. The results of this study suggest that heterogeneous behavior of robots in a multi-human group can play a major role in how group coordination dynamics stabilize (or fail) [77].

- **Developed human-inspired adaptation and anticipation models for robots.** I designed and implemented algorithms for robots to coordinate with the other members of a team in tempo-changing environments, inspired by the neuroscience literature. These algorithms leveraged a human-like understanding of temporal anticipation and adaptation during the coordination process. I validated the algorithms by applying them in a human-robot synchronous drumming scenario. The results suggested that an adaptation algorithm alone enabled a robot to achieve human-level performance in a synchronous drumming scenario. By combining an anticipatory knowledge along with an adaptation process, a robot can perform even better than people during many tempo-changing situations [77].

- **Developed a time-synchronous multi-modal data capture system.** I have identified many of the aspects and challenges of building a time-synchronous data capture system, which is an essential element to investigate group interactions precisely. Based on those findings, I have designed and implemented a time-synchronous, multi-modal data capture system based on client-server architecture to detect high-level human activities [74].

## 1.3 Publications

Some of the work presented in this dissertation is based on the following publications. In addition to these publications, some of the work in this dissertation is also based on papers that have yet to be published.

1. Iqbal, T. and Riek, L. D. (2017). Coordination Dynamics in Multi-human Multi-robot Teams. IEEE Robotics and Automation Letters (RA-L), vol. 2, no. 3, pp. 1712-1717 (Also appears in IEEE International Conference on Robotics and Automation (ICRA)).

2. Iqbal, T. and Riek, L. D. (2017). Human Robot Teaming: Approaches from Joint Action and Dynamical Systems. Humanoid Robotics: A Reference (In review).

3. Iqbal, T. and Riek, L. D. (2017). Can robots synchronize with humans in tempo changing environments?. In Proc. of the 6th Annual Joint Action Meeting (JAM).

4. Iqbal, T., Rack, S., and Riek, L. D. (2016). Movement Coordination in Human Robot Teams: A Dynamical Systems Approach. IEEE Transactions on Robotics (T-RO), vol. 32, no. 4, pp. 909-919.

5. Iqbal, T., Moosaei, M., and Riek, L. D. (2016). Tempo Adaptation and Anticipation Methods for Human-Robot Teams. In Proc. of the Robotics: Science and Systems (RSS), Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics Workshop.

6. Iqbal, T. and Riek, L. D. (2016). Human Coordination Dynamics with Heterogeneous Robots in a Team. In Proc. of the ACM/IEEE International Conference on Human Robot Interaction (HRI) Extended Abstract, pp. 619-620.

7. Iqbal, T. and Riek, L. D. (2016). Leveraging Coordination Dynamics to Enable Human Robot Teams. In Proc. of the Midwest Robotics Workshop (MWRW).

8. Iqbal, T. and Riek, L. D. (2015). A Method for Automatic Detection of Psychomotor Entrainment. IEEE Transactions on Affective Computing, vol. 7, no. 1, pp. 3-16.

9. Iqbal, T. , Gonzales, M. J., and Riek, L. D. (2015). Joint Action Perception to Enable Fluent Human-Robot Teamwork. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 400-406.

10. Iqbal, T. and Riek, L. D. (2015). Detecting and Synthesizing Synchronous Joint Action in Human-Robot Teams. ACM International Conference on Multimodal Interaction (ICMI), pp. 581-585.

11. Iqbal, T., Rack, S., and Riek, L. D. (2015). Will You Join The Dance? Toward Synchronous Joint Action in Human Robot Teams. In Proc. of the 6th Annual Joint Action Meeting (JAM).

12. Rack, S., Iqbal, T., and Riek, L. D. (2015). Enabling Synchronous Joint Action in Human-Robot Teams. In Proc. of the 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI) Extended Abstract, pp. 153-154.

13. Iqbal, T., Gonzales, M. J., and Riek, L. D. (2014). Mobile Robots and Marching Humans: Measuring Synchronous Joint Action While in Motion. AAAI Fall Symposium on Artificial Intelligence in Human-Robot Interaction (AI-HRI), pp. 90-92.

14. Iqbal, T. and Riek, L. D. (2014). Role Distribution in Synchronous Human-Robot Joint Action. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Towards a Framework for Joint Action Workshop.

15. Iqbal, T. and Riek, L. D. (2014). Assessing Group Synchrony During a Rhythmic Social Activity: A Systemic Approach. Conference of the International Society for Gesture Studies (ISGS).

16. Iqbal, T., Gonzales, M. J., and Riek, L. D. (2014). A Model for Time-Synchronized Sensing and Motion to Support Human-Robot Fluency. In Proc. of the 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Workshop on Timing in HRI.

## 1.4 Ethical procedures

Human subjects experiments described in this dissertation were formally reviewed by the Institutional Review Board at the University of Notre Dame. The board found the experiments straightforward to review as they did not involve minors, cause any harm to participants, or result in non-identifiable data being disclosed.

Informed consent was collected from all participants in all experiments. Participants for in-person experiments were compensated for their participation. All participant data was stored securely and was appropriately anonymized.

## 1.5 Dissertation overview

This dissertation is organized as follows:

- **Chapter 2:** Background - a brief overview of robotics, human-robot interaction, and methods to measure the degree of coordination and their relationship in the context of human-robot teaming.
- **Chapter 3:** A time-synchronous data capture system - presents development and validation of a time-synchronous multi-modal data capture system.
- **Chapter 4:** A new method to measure team coordination - presents a novel event-based method to measure the degree of synchronization of a group.
- **Chapter 5:** Validation of the method with a human team - presents validation of the method by applying it to a human group during a rhythmic activity, and discusses a comparison with two other state of the art methods.
- **Chapter 6:** Validation of the method with a human-robot team - presents a validation of the method by applying it to a human-robot team while both the robots

and co-present humans are moving, and explores the application of this method on detecting asynchronous conditions.

- **Chapter 7:** An anticipation method for robot control - presents a new anticipation method that relies on team dynamics to enable a robot to predict events during a human-robot interaction scenario.

- **Chapter 8:** Movement coordination in human-robot teams - presents a study involving an investigation of the movement coordination in human-robot teams by employing the presented anticipation methods on a robot, and discusses the finding of the study.

- **Chapter 9:** Coordination dynamics in multi-human multi-robot teams - investigates the effect of multiple robots (varied by their number and behavior) on the coordination dynamics in multi-human, multi-robot teams.

- **Chapter 10:** Adaptation and anticipation models for robots - presents new algorithms for robots to coordinate with people in tempo-changing environments.

- **Chapter 11:** Conclusion - presents the main contributions of this research, future work, and open questions for the HRI community.

# Chapter 2

# Background

## 2.1 Overview of human-robot teaming

Robotic systems have long been involved in assembly lines automating and increasing efficiency of monotonous or dangerous factory procedures [223, 20]. As technology advances, autonomous robots are becoming more involved in human society in a variety of roles. They are now working alongside humans in teams in many situations, and their presence will be more pronounced in the near future [88].

Similar to human-human teams, robots and humans work together as a team to achieve a common goal in a human-robot team [13]. Based on the goal of the task, a human-robot team may be involved in a task that can be categorized as one of the followings: collaborative, cooperative, or competitive [89, 90].

In a collaborative task, both humans and robots work as equal partners towards a common goal. In a cooperative task, the workload is not equally distributed among human and robot participants. Lastly, in the competitive task, the human and the robot participate in the task as opponents.

In a cooperative human-robot interaction scenario, Ong et al. ([154, 153]) de-

fined five possible role distributions between human and robots to achieve cooperation, and explained it in the context of a telerobotics system. The possible role distributions are master-slave, supervisor-subordinate, partner-partner, teacher-learner, and fully autonomous. To describe high-level interactions between humans and robots, this framework is useful, however it might not be suitable for detailed analysis [90].

In a master-slave model, the human always controls the activities of the robots [154, 153]. The robots only can perform the tasks assigned to it, but the human makes the decision to accomplish the goal.

In the case of the supervisor-subordinate role distribution model, the human acts as a supervisor. The responsibility of the supervisor includes dividing the tasks in smaller subtasks and assigning them to the subordinates. The robots act as subordinates in this model, and have freedom to plan and execute to finish the assigned subtask. If a task goes wrong, it is the responsibility of the supervisor to step in and find a solution of that problem.

Both humans and robots are viewed as partners in the partner-partner role distribution. Here both humans and robots are assigned with some specific tasks, and both can help each other make necessary decisions and actions in the case when a task oriented problem arises. In this scenario, humans and robots have the same level of authority.

In a teacher-learner role distribution, a human acts as the teacher and teaches the robot. It is assumed that the learners will have the ability to understand and learn from the human. A special case of the teacher-learner model is the ‘learning from demonstration’ (LfD) model. In LfD, a learner observes the actions of the teacher, and tries to replicate the actions with or without taking any direct guidance from the teacher [7]. The robots have the ability to take decisions and accomplish tasks without any intervention of human in a fully autonomous role distribution.

Another commonly used role distribution is the ‘leader-follower’ model. In

this model, only the follower adjusts its actions based on the action of the leader [89]. Typically, the human acts as the leader and the robots act as the follower. For example, Stuckler [208] employed a leader-follower based role distribution in their work. In their work, the human and the robot worked together to lift and lower a table. The leader (the human) showed his/her intention of lifting or lowering a table to the follower (the robot). After interpreting the intention of the leader, the follower took appropriate actions to help the leader to lift or lower the table together.

While working alongside humans, a robot might encounter people performing various social actions, such as engaging in social activities, or performing synchronous movements [166]. For example, Ros et al. ([187]) used a humanoid robot to play the role of a dance instructor with children, and Fasola et al. ([44]) designed a socially assistive robot to engage older adults in physical exercise. Others have used robots to dance and play cooperatively with children in therapeutic settings [133, 155]. Koenemann et al. ([104]) demonstrated a system which enabled humanoid robots to imitate complex human whole-body motion.

The following subsections will outline a couple of main application areas where robots cooperatively perform joint action tasks with humans.

### 2.1.1 Activity recognition from robots

To perform contingently with humans in a team, a robot needs the ability to understand the tasks that copresent humans are doing. To address this challenge, researchers from the activity recognition, robotics, and neuroscience communities generally consider psychomotor tasks to be comprised of *motor primitives* (also referred as motor schemas, control modules, and prototypes) [92, 45, 144, 196, 197]. The idea is that psychomotor tasks are comprised of "building blocks at different levels of the motor hierarchy" [144]. Stored primitives are syntactically combined to enable a wide range of complex actions.

While there are theoretical debates in the aforementioned communities about the optimal way to model both primitives and higher-order tasks [194, 11], for practical purposes, researchers have successfully built working systems over the past few decades [3].

*Kinematically-defined primitives* is another approach for understanding human activities, which relate to sequences of movements made by the limbs in 3D space. Many successful approaches have been employed for detecting these primitives [3], with strong results realized for detecting a range of human behaviors from gross motor motion (walking, lifting) to manipulation (stacking objects, brushing teeth) [227, 229, 145, 107, 152, 5, 26].

Ryoo et al. ([193, 192]) discussed various challenges of recognizing interaction-level human activities from a first person view-point. This is very important from robot's point of view, as a robot needs to understand what activity others are performing to it. The authors proposed a new approach to detect interactional activities from continuous videos from a robot, or wearable camera.

Collective activity recognition is another approach to representing the interaction of people in groups. Collective activity refers to coordinated group actions such as queuing in a line. Choi et al. [29, 28] presented a unified framework for tracking multiple people in a crowd while estimating their collective activities. They divided the problem into three parts: detecting individual activities, detecting interactions between pairs, and measuring the collective activities of a group. Chang et al. [24] addressed this problem with a learning-based approach, in which each class of collective activities is modeled based on person-person interaction patterns.

Recently, Neural Networks have been widely used for human activity recognition [93, 40, 96]. Karpathy et al. ([95]) provided an extensive empirical evaluation of Convolutional Neural Networks (CNNs) on large scale video classification using a dataset of 1 million videos. Donahue et al. ([40]) proposed a Long-term Recurrent

Convolutional Networks (LRCNs) architecture, which can take variable length input (e.g., video frames) and can produce a variable length output (e.g., natural language text).

### 2.1.2 Proximate Human-Robot Teaming

In many interactions, robots and humans need to share a common physical space to interact. Various methods are employed on robots to work efficiently in close proximities by avoiding collisions, such as models from a human demonstration, anticipatory action planning, etc. [211].

To build policies for robots to share a space with humans, many approaches in the literature first built models from human demonstrations. After training, robots then use these trained models to collaborate with people. For example, Ben Amor et al. [6] collected human motion trajectories as Dynamic Movement Primitives (DMP) from a human-human task. After that, the authors used dynamic time warping to estimate the robot's DMP parameters. Using these parameters, they modelled human-robot joint physical activities using a new representation, called Interaction Primitives (IP). Their experimental results suggested that a robot successfully completed a joint physical task with a person when IPs were used.

Nikolaidis et al. [148] proposed a two-phase framework to fit a robot's collaborative policy to fit with a human collaborator. They first grouped the human activities into clusters and then learned a reward function for each cluster using an inverse reinforcement learning. This learned model was incorporated with a Mixed Observability Markov Decision Process (MOMDP) policy with the human type as the partially observable variable. After that, they used this model for a robot to infer the human type and to generate the appropriate policies.

Many researchers try to achieve successful human-robot collaboration in a shared space by modeling human activities and use that knowledge as an input to a robot's

anticipatory action planning mechanism [211]. This approach enables robots to generate movement strategies to efficiently collaborate with people.

For instance, Hoffman and Weinberg [70, 68] developed an autonomous robotic jazz-improvising robot, Simon, which played the marimba. To play in real-time with a person, the robot needed an anticipatory action plan. The authors divided the actions into preparation and follow-through steps. Based on the anticipatory plans, their robot could simultaneously perform and react to shared activities with people.

Koppula et al. [108] also developed a method to anticipate a person's future actions. Anticipated actions were then used to plan appropriate actions for a robot to perform collaborative tasks in a shared environment. In their method, they model humans through low-level kinematics and high-level intent, as well as using contextual information. Then, they modeled the human's and robot's behavior through a Markov Decision Process (MDP). Their results suggested that this approach performed better than various baseline methods for collaborative planning.

Mainprice and Berenson [127] presented a framework to allow a human and a robot to perform a manipulation task together in close proximity. This framework used early prediction of the human motion to generate a prediction of human workspace occupancy. Then, they used a motion planner to generate robot trajectories by minimizing a penetration cost in the human workspace occupancy. They validated their framework via simulation of a human-robot collaboration scenario.

Along these lines, Pérez-D'Arpino et al. [157] proposed a data-driven approach which used human motions to predict a target during a reaching-motion task. Unhelkar et al. [214] extended this concept for a human-robot co-navigation task. This model used "human turn signals" during walking as anticipatory indicators of human motion. These indicators were then used to plan motion trajectories for a robot.

### 2.1.3 Human-Robot Handovers

A particular kind of activity often conducted in the proximate human-robot interaction space is a handover. It is an active application space in robotics research [211]. Most of the work on handovers focuses on designing algorithms for robots to successfully hand objects to people, as well as receive objects from them. The researchers working in this area use many methods to achieve their goals, including: nonverbal signal analysis, human-human handover models, and legible trajectory analysis.

Many researchers used non-verbal signals of people to facilitate fluent object handover during human-robot interaction [211]. These signals included eye gaze, body pose, head orientation, etc. For example, Shi et al. [205] focused on building a model for a robot to handover leaflets in a public space, looking specifically at the relationship between gaze, arm extension, and approach. They used a pedestrian detector in their implementation on a small humanoid robot. Their results showed that pedestrians accepted more leaflets from the robot when their approach was employed than another state of the art approach.

Similarly, Grigore et al. [54] demonstrated that the integration of an understanding of joint action into human-robot interaction can significantly improve the success rate of robot-to-human handover tasks. The authors introduced a higher-level cognitive layer which models human behavior in a handover situation. They particularly focused on the inclusion of eye gaze and head orientation into the robot's decision making.

Other researchers also investigated human-human handover scenarios to get inspiration to build models for human-robot handover scenarios [211]. Along this line of research, Huang et al. [73] analyzed data from human dyads performing a common household handover task - unloading a dish rack. They identified two coordination strategies that enabled givers to adapt to receivers' task demands, namely proactive and reactive methods, and implemented these strategies on a robot to perform the same task

in a human-robot team. Their results suggested that neither proactive nor reactive strategy can achieve both better team performance and better user experience. To address this challenge, they developed an adaptive method to achieve a better user experience with an improved team performance compared to the other methods.

To improve the fluency of a robot's actions during a handover task, Cakmak et al. [18] found that the failure to convey an intention of a robot to handover an object causes delay during the handover process. To address this challenge and to achieve fluency, the authors tested two separate approaches on a robot: performing distinct handover poses and performing unambiguous transitions between poses during the handover task. They performed an experiment where a robot used these two approaches while handing over an object to a person. Their findings suggested that unambiguous transition between poses reduced human waiting times, resulted in a smoother object handover. However, distinct handover poses did not have any effect on that.

Other researchers work on perform trajectory analysis to achieve smooth handover of objects. For example, Strabala et al. [207] proposed a coordination structure for human-robot handovers based on human-human handover. The authors first studied how people perform handovers with their partners. From this study, the authors structured how people approach, move their hands, and transfer objects. Taking inspiration from this structure, the authors then developed a similar handover structure for human-robot handover. This human-robot handover structure concerned about what, when and where aspects of handovers. They experimentally validated this design structure.

### 2.1.4 Fluent Human-Robot Teaming

Many researchers in the robotics community try to build fluent human-robot teams. To achieve this goal, many approaches have been taken, including: insights from human-human teams, cognitive modeling for robots, understanding the coordination

dynamics of teams, and adaptive future prediction methods [211].

To achieve fluency in human-robot teams, many researchers investigated how people achieve fluent interaction in human-only teams. This knowledge is used to develop strategies for robots to achieve fluent interaction while interacting with people.

Taking insights from human-human teaming, Shah et al. [203, 204] developed a robot plan execution system, called Chaski, to use in human-robot teams. This system enables a robot to collaboratively execute a shared plan with a person. This system can schedule a robot's action, and adapt to the human teammate to minimize the human's idle time. Through a human-robot teaming experiment, the authors validated that Chaski can reduce a person's idle time by 85%.

To build cognitive models for robots, researchers build on many other fields, including cognitive science, neuroscience, and psychology. For example, Hoffman and Breazeal [63] address the issue of planning and execution through a framework for collaborative activity in human-robot groups by building on the various notions from cognitive science and psychology literature. They presented a hierarchical goal-oriented task execution system. This system integrated human verbal and nonverbal actions, as well as robot nonverbal actions to support the shared activity requirements.

Building adaptive models based on a prediction of future actions is another approach to achieve fluent human-robot collaboration. Hoffman and Breazeal [66] developed a cognitive architecture for robots, taking inspiration from neuropsychological principles of anticipation and perceptual simulation. In this architecture, the fluency in joint action achieved through two processes: 1) anticipation based on a model of repetitive past events, and 2) the modeling of the resulting anticipatory expectation as perceptual simulation. They implemented this architecture on a non-anthropomorphic robotic lamp, which performed a human-robot collaborative task. Their results suggested that the sense of team fluency and the robot's contribution to the fluency significantly

increased when the robotic lamp used their developed architecture.

In other work, Hoffman and Breazeal [65] proposed an adaptive action selection mechanism for a robot in the context of human-robot joint action. This model made anticipatory decisions based on the confidence of their validity and relative risk. They validated their model through a study involving human subjects working with a simulated robot. They used two versions of robotic behaviors during this study, one was fully reactive, and another one used their proposed anticipation model. Their results suggested a significant improvement in best-case task efficiency and significant difference in the perceived commitment of the robot to the team and its contribution to the team's fluency and success.

### 2.1.5 Capabilities that promote robot partnership

There are still many open areas regarding social interactional capabilities that a robot should have before it can fluently and naturally interact with people as a partner. Many researchers have tried to tackle these open questions by building models for robots to understand and to act appropriately as a partner in social situations.

For example, Leite et al. [120] conducted an ethnographic study to investigate how a robot's capability of recognizing and responding empathically can influence an interaction. The authors performed the study in an elementary school where children interacted with a social robot. That robot had the capability of recognizing and responding empathically to some of the children's affective states. The results suggested that the robot's empathic behavior had a positive effect on how children perceived the robot.

Many researchers also explored how a robot's explicit behavior can influence its interaction with people. For example, Riek et al. [178, 181] investigated how imitation by a robot affects human-robot teaming. They designed a study where a robot performed three head gestures while interacting with a person: full head gesture mimicking, partial

mimicking, and no mimicking. The authors found that in many cases people nodded back in response to the robot's nodding during interactions. They suggested incorporating more gestures, along with head nods, while studying affective human-robot teaming.

In another study, Riek et al. [173] explored the effect of cooperative gestures performed by a humanoid robot in a teaming scenario. The authors performed an experiment where they manipulated the gesture type, the gesture style, and the gesture orientation performed by the robot while interacting with people. Their results suggested that people cooperate more quickly when the robot performed abrupt ("robot-like") gestures, and when the robot performed front-oriented gestures. Moreover, the speed of people's ability to decode robot gestures is strongly correlated with their ability to decode human gestures.

In HRI, eye gaze can provide important non-verbal information [211]. For example, Moon et al. [135] performed an experiment where a robot performed human-like gaze behavior during a handover task. In their experiment, a PR2 robot performed three different gaze behaviors while handing over a water bottle to a person. The results indicated that the timing of handover and the perceived quality of the handover event were improved when the robot showed a human-like gaze behavior.

Admoni et al. [1] explored whether a deviation from a robot's standard behavior can influence the interaction. The authors claimed that people often times overlooked robot's standard non-verbal signals (e.g., eye gaze) if they were not related to the primary task. In their experiment, the authors manipulated the handover behavior of a robot to deviate a little from the standard expected behavior. The results of this experiment suggested that a simple manipulation on standard handover timing of a robot made people be more aware of other nonverbal behaviors of the robot, such as eye gaze behavior.

Another well-investigated approach in the field is to teach a robot appropriate behaviors by teaching it through demonstration, i.e., learning from demonstration (LfD)

[7]. For instance, Niekum et al. [146] developed a method to discover semantically grounded primitives during a demonstrated task. From these primitives, the authors then built a finite-state representation of the task. The authors used a Beta Process Autoregressive Hidden Markov Model to automatically segment demonstrations into motion categories. These categories were then further divided into motion grounded stated in a finite automaton. From many demonstrated examples, this model was trained on a robot.

Hayes [60] looked at mutual feedback as an implicit learning mechanism during an LfD scenario. The authors explored grounding sequences as a feedback channel for mutual understanding. In their study, both a person and a robot provided nonverbal feedback to communicate their mutual understanding. The results from the experiments showed that people provided implicit positive and negative feedback to the robot during the interaction, such as by smiling or by averting their gaze from the robot. The results of this work can help us to build adaptable robot policies in the future.

Brys et al. [16] explored how to merge reinforcement learning and LfD approaches together to achieve a better and faster learning phase. One key limitation of reinforcement learning is that it often requires a huge amount of training data to achieve a desirable level of performance. For a LfD approach, there is no guarantee about the quality of the demonstration, which can have many errors. Brys et al. investigated the intersection between these two approaches and tried to speed up the learning phase of RL methods using an approach called reward shaping.

## 2.2 Understanding group interaction dynamics

However, sometimes it can be difficult for a robot to perceive and understand all of the different types of events involved during these activities to make effective

decisions, due to sensor occlusion, unanticipated motion, narrow field of view, etc. On the other hand, if a robot is able to make better sense of its environment and understand the interaction dynamics, then it can make effective decisions about its actions. If the robot has these understandings, then its interactions within the team might reach to a higher-level of coordination, resulting in a *fluent* meshing of actions [180, 76, 64, 18].

Many disciplines have investigated interaction dynamics within teams, which include sociology, psychology, biology, music and dance [140, 141, 121, 122, 30, 198, 156, 100, 111, 62, 114, 61]. For example, Nagy et al. ([140, 141]) investigated collective behavior on animals, and developed automated methods for assessing social dominance and leadership in domestic pigeons. Their investigation explored the effect of social hierarchical structure on dominance and leadership. Their results indicated that dominance and leadership hierarchical structures were independent from each other.

Inspired from bird flocks and fish schools, Leonard et al. ([121, 122]) investigated how collective group motion emerges when basic animal flocking rules (i.e., cohesive and repulsive element) are applied on a group of human dancers. Using tracked trajectories of head positions of individual dancers, the authors developed a time-varying graph-based method to infer conditions under which certain dancers emerged as the leaders of the group.

Khalil et al. [101] explored whether there exists any relationship between a person's ability to synchronize in a group with his/her ability to attend. To investigate that, they designed a musical study where a group of students played five-keyed metallophones while following a leader. During the study, they measured each person's ability to synchronize with the group. After the study, they measured their attention behavior using standard questionnaires. Their results suggested that a child's ability to synchronize with others in a group is an indicator of his/her attention behavior.

Synchronous motion, or joint action, is a common type of high-level behavior

encountered in human-human interaction. It is a naturally present social interaction that occurs when two or more participants coordinate their actions in space and time to make changes to the environment [201]. Understanding synchronous joint action is important, as it helps to accurately understand the affective behavior of a team, and also provides information regarding the group level cohesiveness [172, 218].

Thus, if a robot has the ability to understand the presence of synchronous joint action in a team, then it can use that information to inform its own actions to enable coordinated movement with the team. It also might learn advanced adaptive coordination techniques the human teams use, such as tempo adaptation or cross-training [216]. In next Section, we will describe various methods used in the literature to measure synchrony between systems.

## 2.3 Measuring the degree of synchronization among systems

Synchrony, or entrainment, is described as the spatiotemporal coordination of signals, which results from rhythmic responsiveness to a perceived rhythmic signal [159]. It can occur in many group contexts, such as dance, music, and games, and can be a result of both intentional and unintentional motor coupling during social interaction. *Intentional synchrony* occurs in cooperative group tasks, such as rowing a boat with a team. In contrast, an example of *unintentional synchrony* would be a group of friends spontaneously coordinating their movements when walking together [172].

Clayton et al. ([30]) describe the entrainment phenomena as “the process whereby two rhythmic processes interact with each other in such a way that they adjust towards and eventually ‘lock in’ to a common phase and/or periodicity”, and suggest that two basic components must be involved. First, there must be two or more autonomous

rhythmic processes or oscillators. And second, the rhythmic processes or oscillators must interact. According to Clayton et al., the interaction between the rhythmic processes can be described as entrainment if the system re-establishes the synchronization after any perturbation.

### 2.3.1 Methods for measuring synchrony

Many disciplines have approached the problem of how to assess synchrony, or coordination in a system. These include: physics, neuroscience, psychology, dance, and music [150, 198, 156, 102, 100, 111, 110, 113, 62, 30, 125, 114, 115, 33, 37]. Recently, researchers in computational fields such as social signal processing and robotics have also become interested in this problem [38, 220, 67].

Across all fields, two types of measurement methods exist. The first is best suited for continuous time series data, e.g., recurrence analysis, correlation-based methods, and phase-based approaches. The second type is well-suited for discrete events in time series data, e.g., event-based methods. We will now describe each of these methods.

### 2.3.2 Measuring synchrony in continuous time series data

*Recurrence analysis* is one of the most widely used methods to measure synchrony [34, 125, 226, 186]. This approach is based on trajectory similarities in phase space. First proposed in physics, this analysis method was inspired by coupled dynamic systems. Recurrence analysis assesses how many times the state of a system visits close to a previous state [129].

The graphical representation of recurrence in dynamical systems is called a recurrence plot (RP) [43]. Its central idea is to plot a dot when a state is sufficiently close to any of its previous states. It is also a useful diagnostic tool to quantitatively

analyze recurrences in a dynamic system. Structures and patterns found in an RP are closely linked to the dynamics of the underlying system [129]. The cross-recurrence plot (CRP) is the non-linear bivariate extension of the recurrence plot [129, 130]. Recurrence quantification analysis (RQA) is used to assess and diagnosis complex dynamic systems using RP and CRP [222, 228].

Varni et al. ([218]) used the RP and RQA measures, and presented a system for real-time analysis of nonverbal affective social interaction in a small group. In their experiments, several pairs of violin players were asked to perform while conveying four different emotions. RQA measured the synchronization of the performers' affective behavior. Konvalinka et al. ([106]) also used RQA for measuring the synchronous arousal between performers and observers during a Spanish fire-walking ritual. This synchronous arousal was derived from heart rate dynamics of the active participants and the audience.

*Correlation* is another approach used to evaluate synchrony among continuous time series data [105]. Typically a time-lagged cross-correlation is applied between two time series using a time-window. For example, Boker et al. ([12]) used the cross-correlation method to determine the symmetry building and breaking of body movements in synchronized dancing, and Quian Quiroga et al. ([164]) used it to measure the synchronization between left and right hemisphere rat electroencephalographic (EEG) channels.

*Phase differences* are another well-explored approach in the literature. Typically, phases are defined by Hilbert or wavelet transforms. For example, Richardson et al. ([172]) proposed a method to assess group synchrony by analyzing the phase synchronization of rocking chair movements. A group of six participants rocked in their chairs with their eyes either open or closed, and they used a cluster-phase method to quantify phase synchronization. Néda et al. ([143]) investigated the development of synchronized

clapping in a naturalistic environment. They quantitatively described the phenomena of how asynchronous group applause starts suddenly, and transforms into synchronized clapping.

### 2.3.3 Measuring synchrony in categorical time series data

While the aforementioned methods work well for measuring synchrony in continuous data, there are instances where it may be useful to have methods that work across categorical time series data, which may define discrete events.

Quian Quiroga et al. ([165]) proposed an event synchronization (ES) method for discrete events which can be used for any time series where events can be defined. This method is simple and fast, and the notion of phase is not required. As ES is based on the relative timing of events, it can also determine a leader-follower relationship between two time series, if one exists. The authors applied their method across two sets of time series data - EEG signals from a rat, and intracranial human EEG recordings taken during an epileptic seizure. In both cases, only singular types of events (i.e., local maxima of input signals) were taken into consideration.

Varni et al. ([219]) proposed an extension of this work to measure group synchronization. They authors described a system called I-DJ, which is capable of retrieving music content based on the interaction patterns (i.e., synchronous motion) in a group of dancers. The synchronization of the group, as well as a possible dominant person or a clique in that group, were measured using their proposed method from the dancers' body motion. However, they too only consider a single type of event when measuring group synchronization.

Dale et al. ([35]) presented a cross-recurrence analysis type method for quantifying the relationship between two time series of categorical data (CRQA), and Coco et al. ([31]) recently released an R package which provides an implementation of this and

similar methods.

## 2.4 Chapter summary

This chapter provided a brief introduction to human-robot teaming, team interaction dynamics, methods to detect synchronization of a group, and the motivation behind my work. Robots are expected to become more involved within teams in the future; however, in order to ensure fluent human-robot teaming, they need to be able to understand various aspects of team dynamics, such as synchronous joint action. The next chapter presents a new method to measure the degree of synchronization in a team.

# Chapter 3

# A time-synchronous data capture system

For humans and robots to work together effectively, it is important that both robots understand people and people understand robots. To address this, the concepts of fluency and legibility have become important topics in robotics research [64, 67, 206, 25, 41]. Fluency means that robots convey their motion in a way that is legible (understandable), enabling humans to predict robots' actions, intentions, and goals, and interact fluently with them [64, 41, 103, 18, 36].

One of the most critical elements for achieving fluency is timing. Timing is used as a measure of efficiency for communicating intent and instructions to a robot [32]. Applications such as situated dialogue, music and entertainment, performance characteristics, and expressive gestures, use timing as a measure for efficiency of interaction, influence how users perceive and respond to robots. Similar to the concept of feedback in human computer interaction (HCI) [158, 191], if robots do not act, react, or provide feedback in accordance with human expectations, the interaction has failed.

Synchronizing the actions of a robotic system is particularly important considering

a robot's actions may act as cues for humans. For example, a sociable robotic aide for medical adherence may make some cue to a human so that they take their medicine at a specific time [53], or an entertainment robot may play music and move so that a participant will dance. Similarly, the actions of people may be affected when a robot stops performing one action, and begins performing another.

Timing of actions play a significant role in various social activities performed in human environments, including turn taking, engagement in group tasks, among others [212, 25, 179, 59, 180, 80, 175, 174, 182].

To act accordingly and in a timely fashion with humans, it is important for robotic systems to be able to sense human activities on time using the data received from its attached sensors. Generally a single sensor can only provide limited information about the environment to a robot, whereas multiple sensors can provide information from different viewpoints [225]. As a result, some research has focused on incorporating data from multiple sensors together in order to improve the perception of the environment of a robot [99, 55].

One crucial problem related to timing of robotic systems focuses on temporally synchronizing actions of the system so that appropriate actions and responses occur at the right time. If a human makes a cue that will alter the actions of a robot, all parts of that robot, including its sensors, movement, and activities, should appropriately follow the same timing scheme so that it may perform the next set of actions in a synchronized manner. Otherwise, a robot's actions may occur at the wrong time, data captured using various sensors may represent mismatched timing patterns, and the robot's interaction with a human might be affected.

One challenge, however, is that incorporating this data in a time-synchronized manner may not always be easy. In the case that these sensors are attached directly to the robot, it is likely that such data will be time-synchronous. However, in the case of

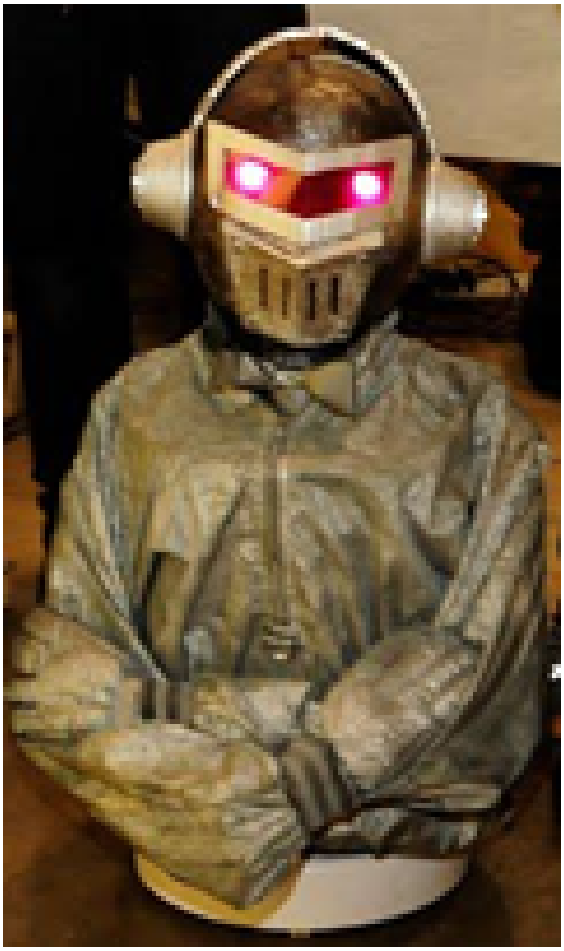

**Figure 3.1**: DJ Rob

a decentralized system of sensors, sensor data are fused locally within a set of local systems, rather than by a central unit, i.e, in a robot [225]. The coordination of sensor data is achieved by communication between the local systems.

Due to the fact that integration of time-synchronous data may directly affect robotic actions, the issue of synchronous data plays a crucial role in the context of HRI. Moreover, in HRI, socially-interactive robots need to operate at a rate at which humans interact, in real-time, in addition to managing synchrony between sensors and data [46]. This makes the issue of time-synchronized data capture and processing much more difficult to manage in HRI.

**Figure 3.2**: Two participants performing in a dance-off

In order to address these issues, we designed a simple client-server based timing model for a robotic system which can handle time-synchronous data in real-time. Our timing model is designed to handle real-time data capture and processing, and to ensure the occurrence of all the actions across our system in a time-synchronous fashion. In addition, our model is not limited to any specific type of sensors, but rather a wide variety of sensors, including video sensors, audio sensors, accelerometers, among others.

We evaluated our model using an entertainment robot, called a DJ Rob, which interacted with dancing participants and measured their expressiveness in real-time (See Figures 3.1 and 3.2). This measure of expressiveness, along with timing, was used to direct our robot to perform specific actions and to provide user feedback. Our results suggest that our model was successful and effective in synchronizing all sensor data, actions, and activities in our system.

## 3.1 System model

Our timing model is designed using a client-server architecture (See Figure 3.3). The server node was responsible for communicating with all client nodes. Each client node was an independent system, and may or may not be connected with its own sensors. A client node may respond to server messages, capture data using its own sensors, process the captured data, and return the processed result to the server. The server node controlled the duration of the data capture and processing of each client system.

We implemented the model using the Robot Operating System (ROS) publisher-subscriber architecture [184]. ROS is an open source platform which provides libraries and tools to develop robotics applications. All nodes in our system run the Electric version of ROS.

In our model, different parts of the model communicate with each other via ROS topics [188]. An ROS topic is a standard unidirectional message passing protocol and uses a publisher/subscriber based model. A publisher can publish messages via an ROS topic, and any number of subscribers can subscribe to that topic. Different parts of the model can communicate with each other by publishing or subscribing to a specific topic.

The server node of our system consists of two modules: the control module (CM), and the decision module (DM). The main task of the CM is to communicate and control the client nodes. This module also communicates with other modules of the server, and is responsible for synchronous data capture and processing in all client nodes. After capturing and processing the sensor data, all client nodes communicate with the DM of the server, and send the processed result to the DM. The decision module can make decisions based on the processed data received from the clients. The decision module may also communicate with robots, controlling their movement.

Each client node consists of a client process module (CPM) and a data capturing

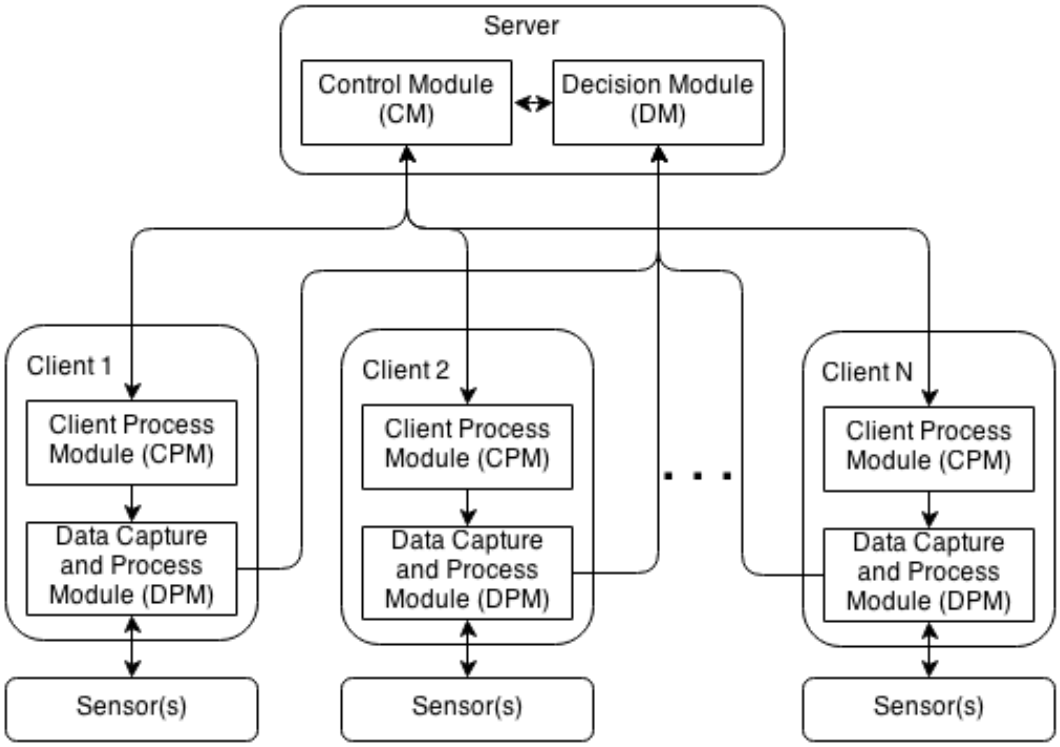


**Figure 3.3**: The architecture of our proposed timing model.

and process module (DPM). The CM of the server node communicates with the CPM module of the client nodes. Based on the message received from the CM, the client process module is responsible for synchronizing the local machine time to a global time server. After synchronizing the time, the CPM is responsible for sending an acknowledgement (ACK) message to let the CM know that the time-synchronization is complete.

These processes also subscribe to the ROS topic broadcasted by the CM. If the CM sends a message to start capturing and processing sensor data, then the CPM communicates with the DPM to make sure that the data capture and processing starts and ends on time. After getting specific message from its own CPM, the DPM starts capturing and processing data for a specific period of time. Following data capture, the DPM publishes the processed data. The DM of the server node and robot nodes

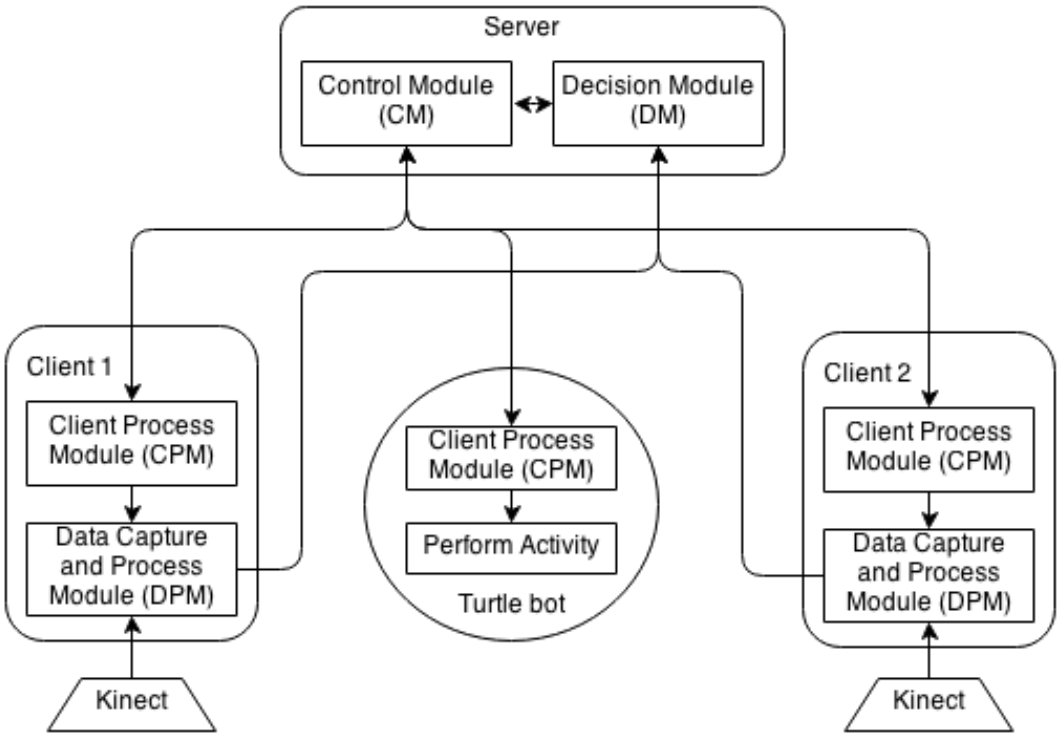


**Figure 3.4**: Implementation of our system. Using our model, each client may either connect to various sensors or robotic agents to perform specific actions.

subscribe to this broadcasted topic from the DPM. The overall system model is presented in Figure 3.3. The arrow represents the direction of communication between modules.

One of the main challenges in synchronized data acquisition is capturing and processing data in a time-synchronized manner. All events that occur at the same time should have the same time stamp in the captured data among all clients. If the events do not represent captured data from each sensor for the same period of time, then it is difficult to make a proper global decision based on the processed data. For example, if event data captured from different sensors are reported from different periods of time, then the decision taken based on the time stamps represents an erroneous result.

In addition, in the case of sensors connected via wired/wireless connections, we encounter the issue of network latency. Depending on the proximity and network congestion, different nodes will receive the broadcasted message at different times.

To overcome these difficulties, we implemented a simple message passing protocol using ROS topics, presented in Figure 3.5. In our protocol, the CM broadcasts a 'sync_time' message. All CPMs of the client nodes and robot nodes subscribe to this topic. Depending on network traffic, different client nodes receive this message at different times.

When a CPM of a client node receives this message, it synchronizes its local time to a global time server using the Network Time Protocol (NTP) [134]. At the same time, the server also synchronizes its time to the global server. Following time synchronization, all CPMs broadcasted an acknowledgement message via an ROS topic. The CM in the server node subscribes to this message and listens for an ACK message from all client nodes to indicate synchronized time in all nodes.

When all client nodes are time-synchronized, the control module of the server broadcasts a 'capture_data' message. This topic consists of three fields. The first field is the 'CR_T' field, which represents the current time of the server system. The second field is the 'STR_AFT' field, which represents a time value in seconds. This field tells the CPM of the client systems to start data capturing and processing after STR_AFT seconds from the CR_T time. Finally, the last field 'DUR' represents the duration for data processing in seconds.

Depending on network traffic, different nodes may receive the 'capture_data' message in different times. The client process module of a client node tells its DPM to start capturing and processing data. All DPMs will wait for a global time to occur, calculated by adding STR_AFT with the CR_T value. As all client nodes are synced with the same global time, all clients begin capturing sensor data at the same time. STR_AFT value should be chosen carefully based on the network congestion, so that all the clients receive the 'capture_data' message within STR_AFT seconds after broadcasted from the server. After the 'DUR' period is over, the DCP module broadcasts the processed result

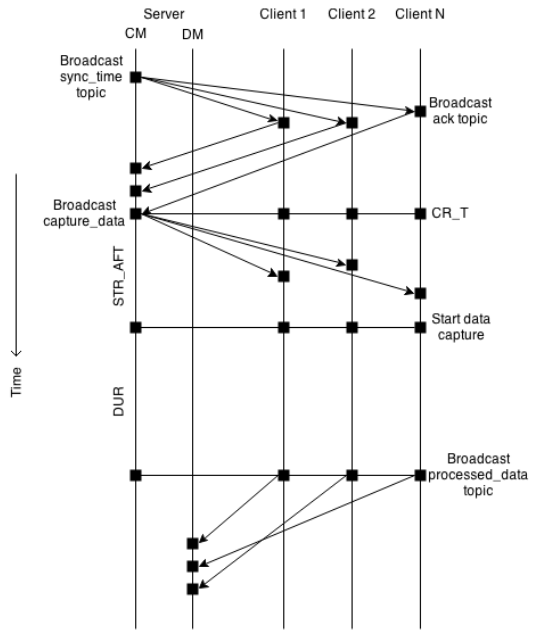


**Figure 3.5**: Message passing protocol of our system

via an ROS topic.

After receiving the processed results from the clients, the DM can make a global decision. Depending on the decision, the DM may instruct the robot connected to the server to act accordingly based on its surroundings, or perform an action.

## 3.2 Model validation

To validate our model, we implemented a system to capture and process RGB-D data from sensors, as well as control robot movements and music in a time-synchronous fashion. This system is composed of three clients and a server: two Microsoft Kinect sensors each connected to a separate computer, a mobile Turtlebot robot (called DJ Rob, who is decorated to look like a dance jockey) [213] who performed specific actions based on pre-defined events, and a stationary system which handled music playback and system actions.

We developed a game in which two participants performed in a dance-off. This dance-off consisted of participants dancing to music for a 60 second duration. During this time, DJ Rob danced along with participants by moving forward and back, allowing his head to bobble, and turning between each participant to simulate judging their dance moves. Figure 3.2 shows two participants performing in a dance-off.

During the dance-off, we measured expressiveness using the Kinect sensors, and processed that data in real-time using our clients. Based on the processed data, the server makes a decision and causes a DJ Rob to perform a specific set of actions based on timing information and RGB-D data (see Figure 3.1). Figure 3.4 shows different parts of our implemented system.

We defined expressiveness as the enthusiasm in the body movements during our study. According to Barakova et al. ([9]), and Lourens et al. ([124]), amplitude and acceleration of the movements of the body joints are sufficient to estimate expressiveness in game settings. Based on these findings, Tetteroo et al. ([210]) used an area of movement per time unit as a measure of expressiveness. To assess quantitatively, we used a similar measure of expressiveness used by Tetteroo et al. ([210]) in our study. We calculated expressiveness by taking the average of the product of speed and distance of body joint movements over time.

Expressiveness is thus defined as:

$$expressiveness = \sum_{time=1}^{T} \sum_{joint=1}^{Joints} (speed \times distance)_{joint}/T;$$

Based on the skeletal information, the DPM module of our system calculates the measure of expressivity. For body joint tracking, we used the OpenNI-based PI_tracker package to perform skeletal tracking of participants using the Kinect sensors. This tracker gives us coordinates and orientations of 15 body joints. Each client node tracked the

skeletal joint position of a participant standing in front of its sensor.

Following each session, each client published their respective expressiveness measure to the decision module of the server node through an ROS topic. Based on the value of expressiveness, the DM determined the more expressive person based on the largest calculated value of expressiveness. The DM then broadcasts the more expressive participant's number, which is processed by DJ Rob, causing it to perform a gesture toward the more expressive participant.

To assess our system model, we conducted a controlled pilot, followed by a real-world experiment with 70 participants (in pairs of two) at a National Robotics Week event held on campus at the University of Notre Dame [142]. Our experiment allowed us to assess the effectiveness of our system model in a real-world environment. For these sessions, all actions made by our robot, as well as music, data capture, and processing are all controlled using our time-synchronous model. These actions are designed to begin at the same time, such that any actions made by dancers are not missed while measuring expressiveness. We captured all data in real-time and processed the result at the end of a one-minute duration to determine the winning action made by DJ Rob.

## 3.3 Results

In general, our overall system model correctly handled all data capture, processing, and actions for our system in a time synchronous manner. In our pilot experiments, we first validated that the robot determined 'no winner' in the case no expressiveness was measured from either client while the music was playing. As the robot determined 'no winner', it did not perform any predefined actions with the participants. We then validated that our robot could determine a winner in the case that a participant only made a small number of movements, while the other did not move. We alternated this level of

expressiveness between the first and second user to ensure that our system successfully captured, processed, and synchronized data with the server, so that the robot could successfully determine the more expressive person. In each of these cases, our robot was successful in performing the correct action.

In addition, our system performed as expected in our real-world experiment. We performed 35 tests, where a pair of participants performed in a dance scenario. During all the tests with our participants, our robot was successful in determining a winner based on the expressiveness measure. To verify that the robot performed the correct action, we manually reviewed each log file. In all cases, our robot correctly moved toward the more expressive person with the higher value of expressiveness reported to the server.

While both clients began capturing data at the same time, we found a small difference in timing between the corresponding frames captured from each device. This is due to the fact that the frames captured from each device may have some small offset from each other due to the device's sampling rate and when it is first powered on. For example, while both clients may start capturing data at time zero and each Kinect has a 20 frames per second (FPS) frame rate, the first Kinect might record its first frame at a slight offset from the second. See Figure 3.6 for an example of this issue.

Considering the frame rate for both Kinects is 20 FPS, each frame is captured on average every 0.05s. Based on our data, the computed average difference between the captured time of the first frames for our two clients across all of our experimental sessions was 0.017 seconds, which is significantly smaller than the time between two consecutive frames.

While we did see a small difference on the millisecond level in our data capture, such a difference will always occur in devices depending on when the first frame occurs, and is negligible. This difference is a result of each sensor's sampling rate, without respect to the time of the client itself. In audio recording devices, for example, such a

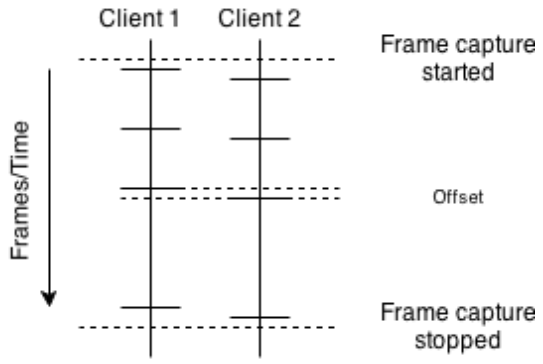


**Figure 3.6**: Capturing data in two clients

difference will be much smaller considering a higher sampling rate, and should not affect synchronization when capturing data. Thus, our results suggest that all data was captured and processed in a time synchronous way across all parts of our system.

## 3.4 Discussion

Timing of actions by both the human and robot is an important factor in HRI. It is important in HRI for robots to perceive human actions and act accordingly in a timely manner. To achieve appropriate action synchronization, it is essential that sensor data and the entire robotic system is time-synchronized. Our proposed simple timing model addresses these issues of real-time data capture and processing from multiple sensors. Our model also ensures the occurrence of all the actions across the system in a time-synchronous fashion.

We anticipate that our model will be useful and beneficial to a number of domains. For example, capturing data from multiple sensors around a room will help a robot have an accurate and up to date picture of the environment around it. In addition, multiple robots could be synchronized in this way to send data from their own respective sensors to a main server to process for a specific time-sensitive task.

Such a model may also be beneficial for researchers in the fields of surveillance or computer vision. Our model may provide an efficient way to keep a global synchronous time space across different parts of a surveillance network. In addition, many computer vision algorithms may take advantage of incorporating data from multiple sensors with camera data for better results using our model.

Furthermore, our model is not limited to specific cameras or sensors. A wide variety of sensors, including audio sensors, accelerometers, among others can be used. Our work gives researchers flexibility in capturing various types of data from a number of sources and fusing them together to obtain a more accurate picture of their surroundings in real-time. Based on this work, we are currently in the process of developing a portable and reusable Robot Operating System (ROS) stack to handle time synchronous actions across all parts of our system interacting with humans.

## 3.5 Chapter summary

In this chapter, I described a simple client-server based timing model to capture multimodal data streams from multiple sensors. This model also included a module to communicate with robots in a time-synchronous manner. I also validated this model with a dance-off scenario, where a robot captured human motion and performed specific actions based on the human's' movements. Since then, I have used this system as a foundation for much of my subsequent work. The next chapter will focus on designing models to understand group behavior from the task-level activities performed by the group members, which later will be extended to implement human-robot teaming scenarios.

## 3.6 Acknowledgements

I thank Michael J. Gonzales for his thoughtful ideas about the experimental design, and also for helping me designing the robot behavior, experiment setup, and performing the experiments. I also thank Ryan McCunn for helping me during the experiment design phase.

This chapter contains material from "A Model for Time-Synchronized Sensing and Motion to Support Human-Robot Fluency," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Workshop on Timing in HRI, 2014 [74]. I was the primary investigator and author of this paper.

# Chapter 4

# A new method to measure team coordination

During a group activity, multiple discrete, task-level events occur, and the outcome and timing of each event depends on the events preceding it. The overall synchronization of the system depends on all these events. Thus, to measure the overall synchronization of a group, we need to model the whole group as a system by incorporating multiple types of events together.

This leads us to explore several research questions. First, can we automatically measure the overall synchronization of a group while taking multiple types of task-level events into account simultaneously? Second, will such an automatic synchronization measure be comparable to group members' own perceptions of synchrony? Third, can such a measure also be used to estimate asynchronous behavior? In this chapter, we seek to address the first research question.

The event based methods proposed in the literature (e.g., CRQA, or event synchronization type methods) work adequately for consecutive, categorical data. However, when the data are sparsely distributed across time (i.e. non-periodic), there are often

instances when no synchronous activity occurs. For example, CRQA includes instances of these "non-events" as actual events while measuring synchrony, which can artificially inflate the true synchrony. Event synchronization type methods do not have this problem; however, they are only able to incorporate a single type of event while assessing synchrony, which is limiting the inherent multimodality of events within human social interaction [109].

Thus, there is a significant gap in this space: there are cases where it may be important to detect synchrony across multiple types of events, and those events may be sparsely distributed. To address this gap, we developed a novel event-based method which can successfully take multiple types of discrete, task-level events into consideration, and successfully ignore the "non-events" while measuring the synchrony of the system. To achieve this goal, we extend the ES method proposed by Quian Quiroga et al. ([165]), as well as the follow-on work by Varni et al. ([219]).

Below, we describe the method to compute each group member's synchronicity to the group as well as the group's overall synchronization, while taking multiple types of events into consideration. First, we will describe the event synchronization method between two time series for one event Later we will extend this method for more than two time series and multiple events.

## 4.1 An event-based synchronization detection method

As described by Quian Quiroga et al. ([165]), suppose $x_n$ and $y_n$ are two time series, where $n = 1,\ldots,N$, and each time series has $N$ samples. Suppose $m_x$ and $m_y$ are the number of events occurring in time series $x$ and $y$ respectively, and $E$ is the set of all events.

The events of both series are denoted by $e^x(i) \in E$ and $e^y(j) \in E$, where, $i =$

$1,\ldots,m_x$, $j = 1,\ldots,m_y$. The event timings on both time series are $t_i^x$ and $t_j^y$ ($i = 1,\ldots,m_x$, $j = 1,\ldots,m_y$) respectively. If the events are synchronous in both time series, then the same event will appear in both more or less simultaneously. Two events are synchronous if the same event appears within a time lag ($\pm\tau$) on both time series.

### 4.1.1 Measuring synchronization of a single type of event across two time series

Now, suppose $c^\tau(x|y)$ denotes the number of times a single type of event $e \in E$ appears in time series $x$ shortly after (within a time lag $\tau$) it appears in time series $y$. Here,

$$c^\tau(x|y) = \sum_i^{m_x} \sum_j^{m_y} J_{ij}^\tau \tag{4.1}$$

Where,

$$J_{ij}^\tau = \begin{cases} 1 & \text{if } 0 < t_i^x - t_j^y < \tau \\ \frac{1}{2} & \text{if } t_i^x = t_j^y \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Similarly, $c^\tau(y|x)$ denotes the number of times a single type of event $e \in E$ appears in time series $y$ shortly after it appears (within a time lag $\tau$) in time series $x$.

$$c^\tau(y|x) = \sum_j^{m_y} \sum_i^{m_x} J_{ji}^\tau \tag{4.3}$$

Where,

$$J_{ji}^\tau = \begin{cases} 1 & \text{if } 0 < t_j^y - t_i^x < \tau \\ \frac{1}{2} & \text{if } t_j^y = t_i^x \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

From $c^\tau(x|y)$ and $c^\tau(y|x)$, we can calculate the synchronization of events in two

time series as,

$$Q_\tau(e) = \frac{c^\tau(x|y) + c^\tau(y|x)}{\sqrt{m_x m_y}} \tag{4.5}$$

$Q_\tau(e)$ represents the synchronization of events in two time series, where we are only considering a single type of event $e$ in both time series. We normalized the value of $Q_\tau(e)$ by the number of events in both time series to get a value between 0 and 1. Thus the value of $Q_\tau(e)$ should be $0 \leq Q_\tau(e) \leq 1$. $Q_\tau(e) = 1$ means that all the events of both time series are fully synchronized. On the other hand, $Q_\tau(e) = 0$ means that the events are not synchronized at all.

$c^\tau(x|y)$ and $c^\tau(y|x)$ values also give us the leader-follower pattern in two time series, if there exists any [165]. This relationship can be incorporated during the calculation of $Q_\tau(e)$ for situations where this pattern might be important.

### 4.1.2 Measuring synchronization of multiple types of events across two time series

$Q_\tau(e)$ gives us the synchronization of events in two time series when only one type of event is considered. In this section, we extend the notion of synchronization of events in two time series for more than one type of event.

Suppose we have $n$ types of events $\{e_1, e_2, \ldots, e_n\} \in E(n)$, where $E(n)$ is the set of all types of events. First, we calculate $Q_\tau(e_i)$ for each event type $e_i \in E(n)$. While calculating $Q_\tau(e_i)$, we will not consider any other event types, except $e_i$. Now, let $m_x(e_i)$ be the number of events of type $e_i$ occurring in time series $x$, and $m_y(e_i)$ is the number of events of type $e_i$ occurring in time series $y$. To measure synchronization of multiple types of events between two time series, we take the average of $Q_\tau(e_i)$, weighted by the number of events of that type. We will call it the synchronization index of that pair. So, the overall synchronization of all events in time series $x$ and time series $y$ is:

$$\forall e_i \in E(n) : Q_\tau^{xy} = \frac{\sum [Q_\tau(e_i) \times [m_x(e_i) + m_y(e_i)]]}{\sum [m_x(e_i) + m_y(e_i)]} \tag{4.6}$$

### 4.1.3 Measuring the individual and overall synchronization index of the group

Using the described method, we will calculate the pair-wise synchronization index for each pair. Suppose we have $H$ number of time series. The time series data are represented as $s_1, s_2, \ldots, s_H$. First, we calculate the pair-wise event synchronization index for each pair. So, we have the value of $Q_\tau^{s_1 s_2}, Q_\tau^{s_1 s_3}, \ldots, Q_\tau^{s_{(H-1)} s_H}$.

Building on the work of Varni et al. ([219]), after calculating the pair-wise synchronization index, we build an undirected weighted graph from these indices, where each time series is represented by a vertex. So, if the time series are $s_1, s_2, \ldots, s_H$, then there is a vertex in the graph which will correspond to a time series. We connect a pair of vertices with a weighted edge, based on their synchronization index value.

There exists an edge connecting two vertices in the graph if the pair-wise synchronization index of the corresponding time series is greater than or equal to a threshold value $Q_{thresh}$. Otherwise, there will be no edge connecting that pair of vertices in the graph. The value of $Q_{thresh}$ should be chosen based on the group task as well as the physical configuration of the group. The weight of that edge will be this pair-wise synchronization index of that pair of vertices. We will refer to this graph as the *group topology graph (GTG)*.

The individual synchronization index depends on both the group composition as well as the size of the group. The size of the group influences the nature of the group in many ways [47]. In some group tasks, an individual may be influenced only by his/her neighbors, whereas in other tasks an individual can be influenced by any member of the group regardless of the group's configuration.

Moreover, the amount of influence may vary based on the size of the group and orientation of the setup. In the case of a very large group, the possibility for each member to be connected with all other members of the group becomes very small [47]. Other group members may have some direct or indirect influences in developing synchrony.

If the group size is small (e.g., four people), we can assume that an individual is influenced directly by all other group members. The individual synchronization index of an individual is measured as the average of the weight of the edges connected to the corresponding vertex in the topology graph. So, the individual synchronization index of series $s_i$ is:

$$I_\tau(s_i) = \frac{\sum_{j=1,\ldots,H,\ j\neq i} Q_\tau^{s_i s_j} \times f(s_i, s_j)}{\sum_{j=1,\ldots,H,\ j\neq i} f(s_i, s_j)} \tag{4.7}$$

Where,

$$f(s_i, s_j) = \begin{cases} 1 & \text{iff } edge(s_i, s_j) \in GTG \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

After calculating the individual synchronization index for each member, the overall group synchronization index is calculated. We take both the individual synchronization index as well as the member's connectivity to the group into consideration while calculating the overall group synchronization index. In a small group, we also consider that each individual is supposed to connect to all other group members in the topology graph when the group is well-synchronized.

For a given vertex in the GTG, the ratio of the number of edges connecting to it, and the number of maximum possible edges in a very synchronized condition for that vertex, is called *the connectivity value* ($CV$). Thus we can define $CV$ of series $s_i$ as:

$$CV(s_i) = \frac{\sum_{j=1,\ldots,H,\ j\neq i} f(s_i, s_j)}{H-1} \tag{4.9}$$

The *CV* represents how well an individual is synchronized with the rest of the group. If an individual is well-synchronized with all other members of the group, then their *CV* value will be 1. On the other hand, if they are not synchronized with any other group members, then their *CV* value will be 0.

While calculating the overall group synchronization index, both the individual synchronization index and the *CV* are taken into account. First, we calculate each individual's synchronization index and multiply it by their *CV*. Then, the overall group synchronization index is computed by taking the average of this product. So, the overall group synchronization index, $G_\tau$, is computed by:

$$G_\tau = \frac{\sum_{i=1,\ldots,H} I_\tau(s_i) \times CV(s_i)}{H} \tag{4.10}$$

## 4.2 Chapter summary

In this chapter, I discussed the development process of our new event-based method to measure the synchronization of a group by taking multiple types of task-level events into account. This method provides us a basis to measure the synchronization of human-human and human-robot groups when the group members are involved in a synchronization task. In next chapter, we will validate our method by applying to a human-human study. We will also compare the results with two other existing methods from the literature during this validation process.

## 4.3 Acknowledgements

I thank James Delaney, Giovanna Varni, Moreno Coco, and Maria O'Connor for their valuable suggestions during the algorithm development phase.

This chapter contains material from "Assessing group synchrony during a rhythmic social activity: A systemic approach," by T. Iqbal and L. D. Riek, which appears in Proceedings of the Conference of the International Society for Gesture Studies (ISGS), 2014 [80], and "A Method for Automatic Detection of Psychomotor Entrainment," by T. Iqbal and L. D. Riek, which appears in IEEE Transaction on Affective Computing, 2015 [82]. I was the primary investigator and author of these papers.

# Chapter 5

# Validation of the method with a human team

In the previous chapter, we presented a new method to measure the overall synchronization of a group while taking multiple types of task-level events together. Now, in this chapter, we are interested to explore whether the synchronization measure described in Chapter 4 will be comparable to group members' own perceptions of synchrony.

We explore this research question by validating the method in three ways. First, in Section 5.1, we validated the method by applying it to a multiple event-based rhythmic game, where each player performs a periodic psychomotor activity within the group, and each player's movements influence all others' movements. Our method was able to model both synchronous and asynchronous activity, and was well-matched to the players' perceptions of synchrony. Second, in Section 5.2, we compared our results to another method from the literature that uses single event types, and found our multiple event-based method to be more accurate in estimating synchrony. Third, in Section 5.3, we compared our results to the cross-recurrence quantification analysis (CRQA) method.


**Figure 5.1**: Game phases during one iteration of the cup game[1]. Game phases are: *c* - clapping, *t* - tapping, *h* - holding, *m* - moving, and *p* - passing the cup. The game is in sequence from left to right, and top to bottom.

The results again suggested that our multiple event-based method is more accurate in estimating synchrony.

In this chapter, we aim to address this research question by validating our method.

## 5.1 Validation of our method applied to a synchronous psychomotor task

To validate our method and address our research questions, we first sought to analyze a group event where every member participateed in a collective task, thus contributing to the overall group synchrony. Thus, we began by analyzing participants playing a tabletop game called "the cup game". This is a cooperative, rhythmic game, played by multiple players who are situated in a circle, and consists of clapping, tapping, moving and passing cups (See Fig. 5.1). It was recently popularized in the Hollywood film *Pitch Perfect*.

In the cup game, each player must exhibit coordinated psychomotor skills, conduct a specific activity at a specific time, and synchronize his/her activity with the group. Therefore, this game serves as a strong testbed for validating our method, and also enables us to explore how group synchronization emerges over time.

In terms of setup, each player stands or sits at a table (or on the floor), and plays the game with their hands and a cup. During every iteration of the game, each player

performs a sequential and rhythmic activity with the cup, and ends the iteration by passing their cup to their neighbor. To maintain the overall rhythm of the game, all participants need to perform their tasks more or less synchronously over time. We will consider all of these activities performed by each player as the *events* of the game.

From a high level, different events during the game can be classified into five categories: clapping ($c$), tapping the cup ($t$), holding the cup ($h$), moving the cup ($m$), and passing the cup ($p$). Fig. 5.1 shows a single iteration of the game; all players perform the following tasks in sequential order: $c-c-t-t-t-c-h-m; c-h-m-m-m-h-p-p$.

We ran a series of experiments where participants played two games in groups of four, for approximately two minutes per game. Two synchronized Kinect sensors recorded RGB and skeletal joint information of the participants during a game[74]. After the games, each participant rated on a discrete visual analog scale how well-synchronized they felt each game was, and which game was more synchronous.

In a highly synchronized game, all game events will happen more synchronously over time. On the other hand, in non-synchronous games, events may happen less synchronously. We hypothesized that each player's perception of game synchrony would be based on the relative timing of game events. (i.e., for games where events are well-timed, players would perceive the game to be more synchronous than those with poorly-timed events). We were also interested to see whether our system's automatic measurement of group synchrony would match player's perceptions.

Thus, in this work, we took players' post-game assessments to be the ground truth upon which our later validations are based. While measuring ground truth using external observers is a reasonable approach in other affective labeling work we do (c.f. [59, 177]), on this project, synchrony seemed better measured using a self-assessment approach. This was both to enable capturing immediacy of playing the cup game, as

the tactile and auditory sensations, which help players feel “in-sync” are challenging if not impossible to replicate for external observers. Furthermore, the literature suggests self-assessment is reliable for data collected over short time periods, as this data was [189].

### 5.1.1 Participants

A total of 22 people participated in our experiment, 50% female. Their average age was 24.8 years old (s.d. = 3.97), and the majority were undergraduate and graduate students. In total, there were six experimental sessions consisting of four players each. Participants were randomly categorized into six groups (two people participated twice).

Participants were trained in how to play the game before the experiment began, including playing one practice game as a group. They then played two games that were recorded. Following the experiment, participants completed a short questionnaire asking them to rate which of the two games they felt was more sychronous.

### 5.1.2 Data collection

Fig. 5.2-A shows the data collection setup. Four players stood around a table to play the game, two on each side. Two Kinect sensors were positioned approximately 86 inches above the ground and 28 inches from the table edge. The sensors tracked RGB and depth information, which afforded the ability to track the body joints of all players and the red-colored cups.

Before the game began, participants performed a brief sensor calibration process. The players stood in front of the Kinect for around 5 seconds to calibrate the sensor. After the calibration process, each Kinect tracked 15 body joint positions for two players. Fig. 5.2-C shows an overview of the system architecture.

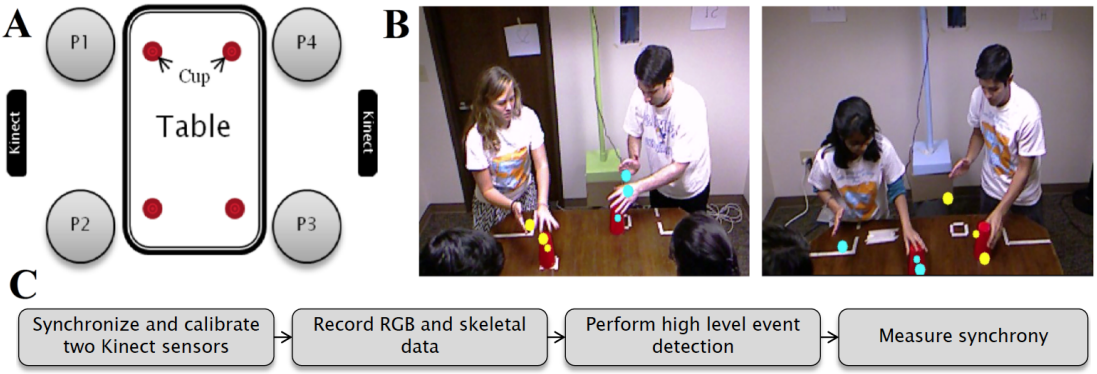


**Figure 5.2**: A) Block diagram of the setup. $P_1$, $P_2$, $P_3$, and $P_4$ refer to Players 1, 2, 3, and 4 respectively, B) Four players playing the cup game. The players' movements are tracked by two synchronized Kinect sensors. The small solid circles represent the center of the cup, and the large solid circles represent the projected 3-D hand joint positions on the RGB image plane. C) High level system architecture.

The sensors were connected to two computers running Ubuntu. Both machines ran the Robot Operating System (ROS) Electric release. ROS is an open source platform, which provides libraries and tools to develop robotics applications. Before data recording began, both systems were synchronized with an Ubuntu time-server to ensure they were accurately keeping time. Data were stored in the *rosbag* file format, which includes timestamps for sensor readings.

### 5.1.3 High level event detection

After completing the data collection, we labeled different steps of the game as the aforementioned high level events ($c$, $m$, and $p$). If any of the hand positions of a player were poorly tracked in a given frame, then we excluded that frame from analysis for that player.

Exclusion was based on the tracking accuracy of the Kinect sensors. Due to hand occlusion at some stages of cup game, the Kinect sensor can not track hand joint positions with full confidence. Therefore, we excluded these frames while performing event detection. On average, we excluded around 35% of frames for a session across

the whole group; in the best case we excluded 12% of frames. However, this frame exclusion rarely interfered with our event detection as it was well-distributed across the sessions. Furthermore, we manually validated the accuracy of each event class before employing our event detection methods. This was performed via human labeling using a representative sample of each event class.

*Cup tracking*

Cups were tracked using standard blob tracking techniques in each frame. As only red color cups were used, the red blobs were tracked from the RGB image using the ROS cmvision package. After discarding very small blobs as noise, the rest of the blobs were considered as candidates for the cup's position. An undirected graph was generated using each blob center as a vertex. Two vertices were connected in the graph if they were closer than a threshold value.

From this resultant graph, we then calculated the connected components. All of the blobs in a connected component were clustered together. The center of each cluster was calculated as the mean position of the blob centers of that cluster, weighted by the area of each blob. Each cluster center was a cup center candidate, and those closer to the hands of the players were considered as the cup positions during the game.

*Clap (c) event detection*

A clap event was detected when the hand joints from the skeletal data were closer to one another than a threshold distance. While calculating the distance between hand joint positions, only the $x$ and $y$ coordinates from the 3D skeletal position were used.

A clap event happened when: 1) the hand joints' distance was within a threshold, 2) the distance between hand joints reached a local minima, and, 3) none of the hand joints were closer to the cup position in the RGB image, as calculated by projecting each 3D hand joint position on the RGB image plane. This helped distinguish clapping events

from tapping events. Thus, clap events only lasted for one frame.

*Move (m) and Pass (p) event detection*

Cups move at several points during a game iteration, note the *m* or *p* steps in Fig. 5.1. If a player moved the cup with their right hand, this was denoted as a move event (*m* in Fig. 5.1), and with their left hand, a pass event (*p* in Fig. 5.1).

If a cup position was changed from the previous frame, then it meant that the cup had been moved by a player. In this case, our system assumed that the player closest to the cup's position moved the cup. To determine the hand positions of each player in the RGB image, the 3-dimensional hand joint positions from the skeletal data were projected into the RGB image plane. Then, in RGB coordinates, our system calculated the distances of the hand positions from the cup's center. If the player's right hand was closer to the moving cup than their left hand, then our system assumed that the player was moving the cup with his/her right hand, and the event was denoted as a move event (*m*) for that player, but if their left hand was closer, we assumed they passed the cup (*p*).

As move and pass events may last for several frames in the data, instead of denoting a move or pass event for every frame, we considered the starting and the ending frame of each event sequence as a new event. The frame when a move event started was denoted as a 'move start' ($m_s$) event, and when the move ended, that frame was denoted as a 'move end' ($m_e$) event for that player. Similarly, for pass event sequences, two events were generated, a 'pass start' ($p_s$) and a 'pass end' ($p_e$) event.

To demonstrate what that looks like, suppose for a few frames our event detector detected move events as follows:

| $Time\,stamps$ : | 1.1s | 1.2s | 1.3s | 1.4s | 1.5s |
|---|---|---|---|---|---|
| $Events$ : | - | $m$ | $m$ | $m$ | - |

This means that a move event started at timestamp 1.2 $s$, continued for one frame, and ended at timestamp 1.4 $s$. From this move event sequence, our system will generate two new events, $m_s$ and $m_e$. A move start event happened when the move sequence started. When the move sequence ended, our system will label that as a move end event. Thus, our processed events would be:

| | | | | | |
|---|---|---|---|---|---|
| *Time stamps* : | 1.1s | 1.2s | 1.3s | 1.4s | 1.5s |
| *Raw events* : | - | $m$ | $m$ | $m$ | - |
| *Processed events* : | - | $m_s$ | - | $m_e$ | - |

### 5.1.4 Overall synchrony detection

After detecting the high level events for each player, each player's data were represented by a time series. Frames received from both Kinects were adjusted to represent the same frame across all time series data. This yielded four time series, each of which represented the high level events associated per player.

For these time series data, the individual and overall synchronization indices of the system were calculated using our method. For an example calculation, suppose the time series were $s_1$, $s_2$, $s_3$, and $s_4$. From these time series, pair-wise synchronization indices $(Q_\tau^{s_1 s_2}, Q_\tau^{s_1 s_3}, \ldots Q_\tau^{s_3 s_4})$ were calculated.

The pair-wise synchronization indices measure the degree to which the events are happening synchronously in both time series. In this case, these two time series are isolated from the rest of the time series of that system. Thus, this measure gives us the notion of how synchronous two players are among themselves when they are isolated from the rest of the group. For example, a higher pair-wise synchronization index may be observed if two players have the continuous tendency to synchronize their events to each other over time. We may observe a higher pair-wise synchronization index value

**Table 5.1**: Individual and Group Synchronization Indices for All Sessions

| Sessions | Games | Indi. Sync. Indices* | | | | Group Sync. Index | | MoP◇ | Agreement(%)§ |
|---|---|---|---|---|---|---|---|---|---|
| | | $P_1$ | $P_2$ | $P_3$ | $P_4$ | GSI value† | s.d.‡ | | |
| $Session_1$ | 1 | 0.51 | 0.53 | 0.50 | 0.52 | **0.51** | 0.05 | 6.72 | **100%** |
| | 2 | 0.48 | 0.53 | 0.47 | 0.51 | 0.49 | 0.04 | | |
| $Session_2$ | 1 | 0.56 | 0.55 | 0.53 | 0.54 | **0.54** | 0.03 | 22.38 | **75%** |
| | 2 | 0.51 | 0.55 | 0.48 | 0.51 | 0.49 | 0.04 | | |
| $Session_3$ | 1 | 0.43 | 0.49 | 0.48 | 0.48 | 0.44 | 0.18 | -4.79 | **100%** |
| | 2 | 0.52 | 0.50 | 0.50 | 0.48 | **0.48** | 0.05 | | |
| $Session_4$ | 1 | 0.32 | 0.40 | 0.38 | 0.44 | 0.29 | 0.13 | -41.94 | **75%** |
| | 2 | 0.49 | 0.57 | 0.52 | 0.56 | **0.54** | 0.03 | | |
| $Session_5$ | 1 | 0.46 | 0.55 | 0.48 | 0.56 | 0.47 | 0.05 | -19.19 | **100%** |
| | 2 | 0.45 | 0.56 | 0.51 | 0.57 | **0.52** | 0.03 | | |
| $Session_6$ | 1 | 0.49 | 0.53 | 0.45 | 0.54 | 0.47 | 0.07 | -7.80 | **75%** |
| | 2 | 0.49 | 0.54 | 0.46 | 0.55 | **0.50** | 0.05 | | |

* Mean value of individual synchronization indices of four players ($P_1$ to $P_4$).
† Mean value of group synchronization indices. For each session, the higher group synchronization index value of the game is highlighted in bold.
‡ Standard deviation (s.d.) of group synchronization indices for each game. A lower s.d. value reflects stronger, or more stable, synchrony, and a higher value reflects weaker, or less stable, synchrony.
◇ Measurement of precision for each *Session* in standard deviations.
§ Percentage of players in each session for which our automated measure produced a match with the players' perception about both games. For example, in $Session_1$ our method produced group sync. indices of 0.51 and 0.49 for Games 1 and 2 respectively. For this session, all four players agreed that Game 1 was more synchronous than Game 2.

if the participants previously played the game together, and they may synchronize with each other more easily.

After calculating these values, the topology graph was generated from the pairwise synchronization indices. From the topology graph, each player's synchronization index $I_\tau(s_i)$ was calculated. This individual synchronization index yielded how well each player was synchronized with rest of the group.

As the group size was small and all of the players were in close proximity, all players influenced one another. If a player made a mistake at any stage of the game, not only did it affect the overall synchrony of the group, but it also affected other individuals' synchrony with the group. From the individual synchronization indices, the overall synchronization of the group was calculated using Equation 8.2.

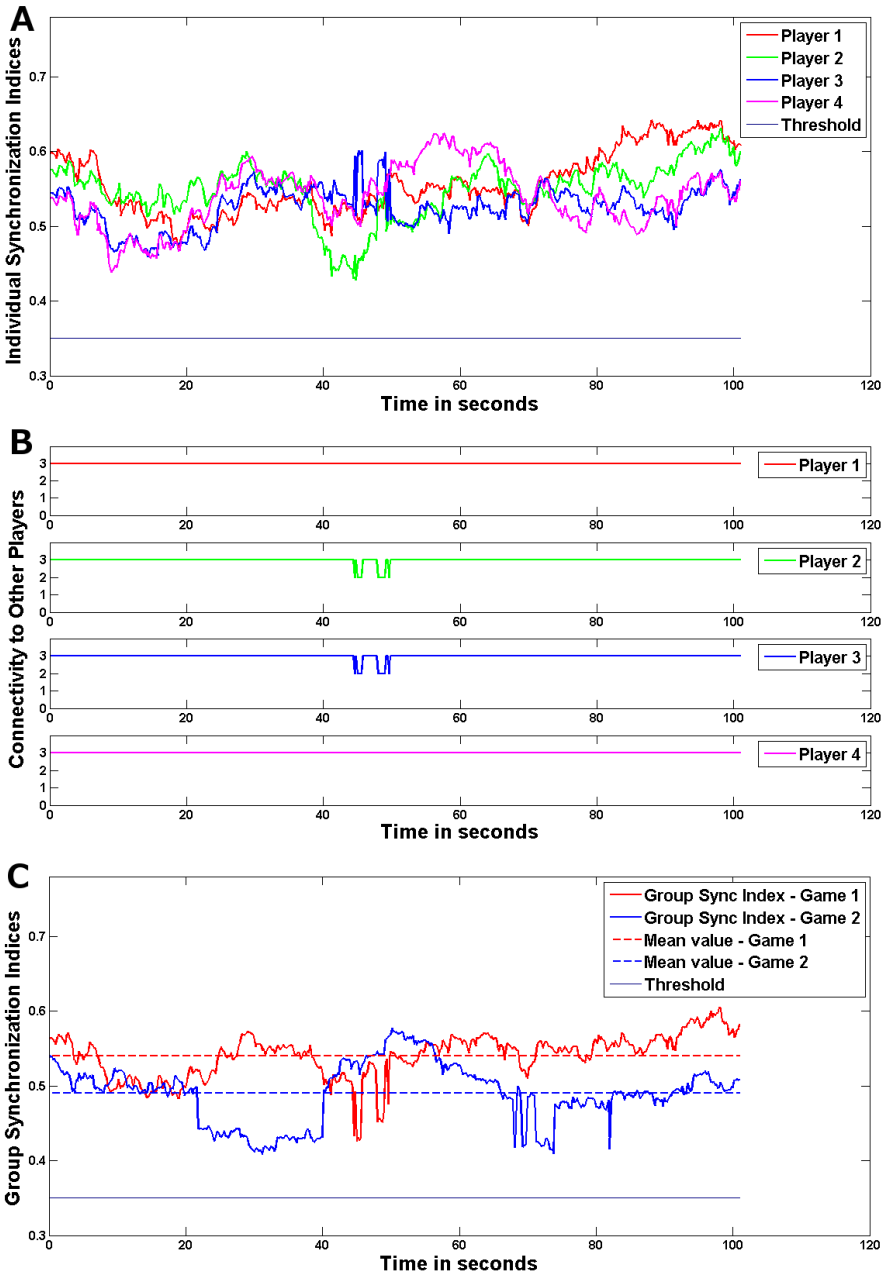


**Figure 5.3**: A) Time vs. Individual Synchronization Indices of four players of $Session_2$, $Game_1$. B) Time vs. Connectivity to Other Players of $Session_2$, $Game_1$ (Player 1 to Player 4, from top to bottom). A Connectivity value of 3 suggests the group was well connected; 0 means no connectivity. C) Time vs. Group Synchronization Indices of both games of $Session_2$.

### 5.1.5 Results

We conducted a total of ten experimental sessions. However, four sessions were excluded due to technical malfunctions (calibration or tracking errors), so here we report

the results from six sessions. A session is defined as a group of four players participating in two games, where each game lasts approximately two minutes. We used a sliding window of 20 $s$ for calculating the pair-wise, individual, and group synchronization indices, as this is approximately how long it took to complete one iteration of the game.

Table 5.1 shows the individual and group synchronization indices from all sessions and games. Here, the values of the synchronization indices are averaged over the duration of each game. The last column of the table represents the concordance between players' perception and our method's perception of which game was more synchronous. For example, for $Session_3$, the group synchronization indices produced by our method are 0.44 and 0.48 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous; thus indicating 100% agreement.

Rather than go into depth for each game, we will now focus on one game in detail, $Session_2, Game_1$. This analysis method is identical for each game, and all games showed similar characters, so the following results reporting is generalizable to all games.

We present the individual synchronization index of four players of one game in Fig. 5.3-A. As we used a sliding window of $20s$, the values present at time $0s$ actually represent the values calculated from the time window $0s$ to $20s$. We used $\tau = 0.21s$ for our calculation, and $Q_{thresh} = 0.35$ as the threshold of synchronization index to generate the topology graph. Based on the data from our representative sample, we found that two people were not synchronous when their pair-wise synchronization index fell below 0.35. Thus, we used this as the threshold in our experiment.

An example of the connectivity of the nodes in the topology graph is shown in Fig. 5.3-B. This contains four sub-graphs, each showing the connectivity of a player over time to other nodes (players) in the graph. Each sub-graph shows the connectivity of one player, with all other players in the topology graph. Time is plotted along the $x$-axis, and the number of connected nodes are plotted along the $y$-axis.

Each individual synchronization index depends on that player's connectivity with others in the topology graph, i.e., the player's synchrony with the other players. From Fig. 5.3-B, one can see that Player 2 is connected with two more nodes in the topology graph in-between the time window of 45 *s* to 50 *s*. This means that one of the pair-wise synchronization index values of $P_1 - P_2$, $P_2 - P_3$, or $P_2 - P_4$ was less than the threshold value. This is also the case for Player 3. This observation means that the pair-wise synchronization index of Player 2 and Player 3 ($P_2 - P_3$) had fallen below the threshold during that period. Thus, there was no edge between that pair in the topology graph during that time window.

As a result, in Fig. 5.3-A, one can see that the individual synchronization index for Player 2 and Player 3 was decreasing before they became disconnected in the topology graph. However, after they were disconnected in the topology graph, their individual synchronization index started to increase. We can explain this situation using an example. Suppose a time window pair-wise synchronization indices of $P_1 - P_2$, $P_1 - P_3$, and $P_1 - P_4$ are 0.5, 0.5, and 0.4 respectively. As the $Q_{thresh} = 0.35$, $P_1$ is connected with all other players in the topology graph.

Given this scenario, $P_1$'s individual synchronization index is $(0.5+0.5+0.4)/3 = 0.47$. Now, assume that in the next time window these values have been changed to 0.5, 0.5, and 0.3 respectively. In the changed scenario, the pair-wise synchronization index value of $P_1 - P_4$ is less than the threshold. Thus, there will not be any edge connecting these nodes in the topology graph. For this window, the individual synchronization index of $P_1$ will be $(0.5+0.5)/2 = 0.5$. Although it is not synchronous with respect to each player in the group, for ones with which it is synchronous, the level of synchrony is relatively higher in degree. One can see this phenomenon in Fig. 5.3-A, just after the connectivity between $P_2 - P_3$ was lost, the individual synchronization indices of both $P_2$ and $P_3$ increased.

For each game of a session, the group synchronization index is calculated. For example, we display group synchronization indices of this session (both games of $Session_2$) in Fig. 5.3-C. In this figure, time in seconds is shown along the $x$-axis, and group synchronization index is shown along the $y$-axis. We calculate group synchronization index for each time window of size 20 $s$. This graph also presents the mean value of group synchronization indices.

The group synchronization index depends on the individual synchronization index and the topology graph. One can see from Fig. 5.3-C, the group synchronization index of $Game_1$ drops between the time window of 45 $s$ to 50 $s$. During the calculation of the group synchronization from the individual synchronization index, we also take each node's 'connectivity value' into consideration. This also supports the fact that the individual synchrony drops and pair-wise synchrony breaks during that period, see Fig. 5.3-A and B.

### 5.1.6 Discussion

From Table 5.1, one can find the percentage of the players' perception for which our measure produced a match for each session. For example, from the table one can see that the majority of participants agreed as a group that $Game_1$ was more synchronous than $Game_2$ in $Session_1$. On the other hand, for $Session_4$, the majority of participants agreed as a group that $Game_2$ was more synchronous than $Game_1$. For all of the sessions, the group synchronization indices produced by our method agreed with the perception of the majority of participants in 100% of the sessions (6 out of 6 sessions).

From the participants' perception, and from our automated method, one can see that the $Game_2$ is more synchronous than $Game_1$ for the last four sessions. One might suspect that this pattern may be attributable to learning effects. After the first game, the players may become used to the rhythm of the game, and learned how to be synchronous

**Table 5.2**: Group Synchronization Indices for All Sessions Computed Using The Three Comparison Methods

| Sessions | Games | Our method | | By Varni et al. | | CRQA |
|---|---|---|---|---|---|---|
| | | GSI † | s.d.‡ | GSI | s.d. | GSI |
| $Session_1$ | 1 | **0.51** | 0.05 | **0.72** | 0.02 | **0.19** |
| | 2 | 0.49 | 0.04 | 0.70 | 0.03 | 0.17 |
| $Session_2$ | 1 | **0.54** | 0.03 | **0.74** | 0.03 | **0.18** |
| | 2 | 0.49 | 0.04 | 0.74 | 0.03 | 0.17 |
| $Session_3$ | 1 | 0.44 | 0.18 | **0.70** | 0.03 | 0.17 |
| | 2 | **0.48** | 0.05 | 0.67 | 0.03 | **0.18** |
| $Session_4$ | 1 | 0.29 | 0.13 | 0.61 | 0.04 | 0.15 |
| | 2 | **0.54** | 0.03 | **0.68** | 0.02 | **0.20** |
| $Session_5$ | 1 | 0.47 | 0.05 | 0.69 | 0.03 | 0.18 |
| | 2 | **0.52** | 0.03 | **0.72** | 0.02 | **0.19** |
| $Session_6$ | 1 | 0.47 | 0.07 | **0.69** | 0.03 | **0.18** |
| | 2 | **0.50** | 0.05 | 0.64 | 0.04 | 0.17 |

† Mean value of group synchronization indices. For each session, the higher group synchronization index value of the game is highlighted in bold.

‡ Standard deviation of group synchronization indices for each game. A lower s.d. value reflects stronger, or more stable, synchrony, and a higher value reflects weaker, or less stable, synchrony.

as a team. Therefore, they showed a higher degree of entrainment during the second game. One also can see that most of the individual synchronization indices show higher values during the second game.

From the results, however, one can see that the first two sessions do not agree with this assumption. This may be due to the fact that most of the participants of the first two sessions had prior experience playing the game. This may explain why a learning effect was not visible during these sessions. Regardless, our automatic method still successfully measured synchrony of these two sessions, which matched with the perception of the players.

Table 5.1 also presents the standard deviation (s.d.) for the group synchronization indices for every game. A lower s.d. value reflects stronger or more stable synchrony, and a higher value reflects weaker or less stable synchrony. For example, the s.d. of the

group synchronization index is less for $Game_1$ than $Game_2$ during $Session_2$. During the other five sessions, the s.d. values of $Game_2$ are less than the values of $Game_1$. This means that $Game_1$ exhibits more stable synchrony than $Game_2$ for $Session_2$. During the other five sessions, $Game_2$ exhibits more stable group synchrony than $Game_1$. These s.d. indices are also aligned with the answers of the participants as a group in 83.33% cases (5 out of 6 sessions).

Table 5.1 also presents the measurement of precision calculation for each *Session*. The measures produced by our method for two games of a session are different by at least an order of magnitude of the standard deviation. Although it appears that the two games only differ slightly in their synchronization indices, the differences are in fact significant.

In Fig 5.4-left, we present the confusion matrix for our method. The actual game is identified from the perception of the majority of the participants for each session. The predicted game for each session is the more synchronous game of that session measured by our method. From this matrix, one can see the the accuracy of our method is 1 (which is presented as 100% in the matrix), precision value for $Game_1$ is 1, precision value for $Game_2$ is 1, recall value for $Game_1$ is 1, and recall value for $Game_2$ is 1.

*Measurement of Precision Calculation*

As the participants played two games during a *Session*, it is important to measure whether the difference in the group synchronization indices measured by our method is within the measurement precision. In this Section, we describe this measurement process in detail.

To calculate the group synchronization index (GSI), we used a sliding window of 20 *s* during each game. We calculated the GSI for each window, and presented the mean value of all windows in Column GSI in Table 5.1. We moved the window by increments of 0.2 *s*.

Now, suppose we have $n$ number of windows during a game, and let us assume

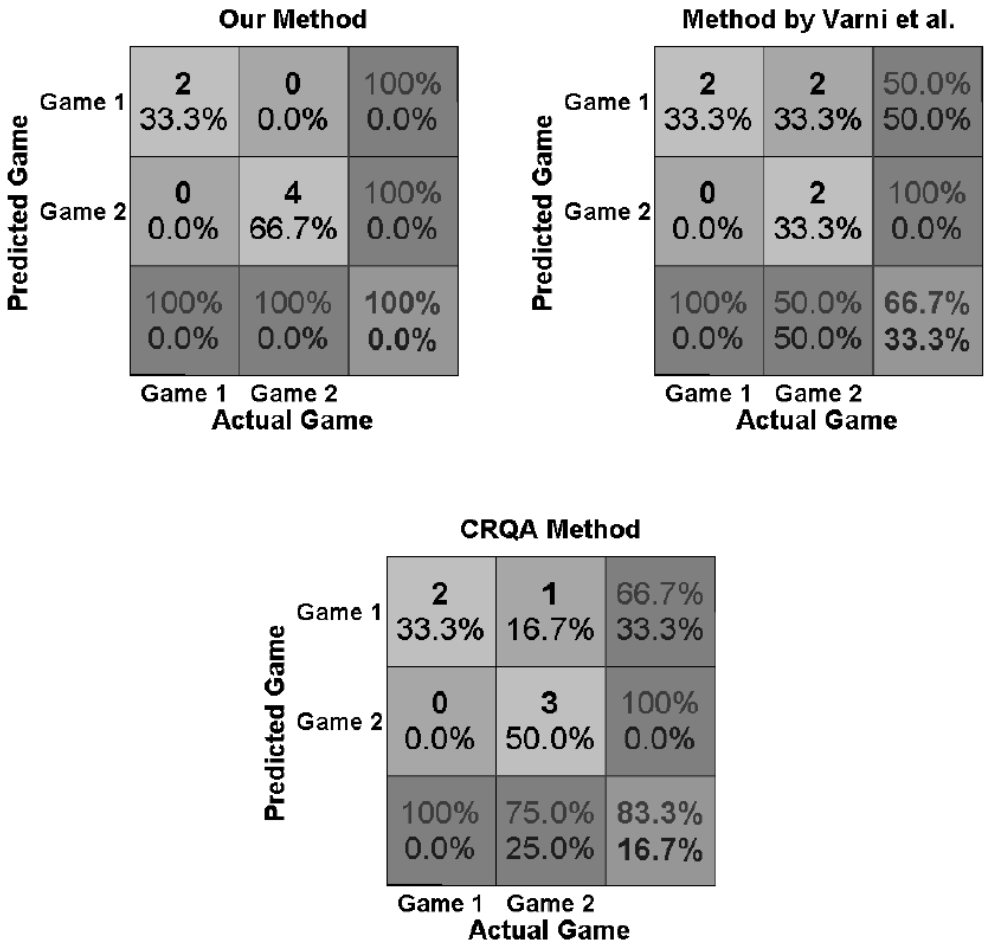


**Figure 5.4**: Confusion matrices for three validation methods. During a session, actual game is identified based on group members' agreement, and the more synchronous game is represented as the predicted game.

that the time series is second order stationary. Let, $X_i$ be a *GSI* value for a window *i*. When, $X_1, X_2, \ldots, X_n$ are independent with a common mean and a variance, $\mu_X$ and $\sigma^2$ respectively, then, $\bar{X}_n$ will have mean $\mu_X$ and variance $(\sigma^2)/n$.

As we are using overlapping windows, $X_i$ is the same as an 'overlapping batch mean' described in Alexopoulos et al. [4]. Here, $S^2$ is approximately the same as the overlapping batch means estimator (O(b,m)). Thus, $S^2$ is a slightly biased but good estimator of $\sigma^2$, the variance of each observed GSI presented in Table 5.1. Now,

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n) \tag{5.1}$$

$S^2/n$ is used to estimate $Var(\bar{X}) = \sigma^2/n$. Using the plug-in principle, we have:

$$Var(\bar{X}_n) \approx S^2/n \tag{5.2}$$

Suppose, for the other game in the session, we have a mean synchronization index value, $\bar{Y}_n$. Now, there will be a significant difference between the quantity $\bar{X}_n$ and $\bar{Y}_n$, if they are different beyond the precision of measurement. Thus, here, the precision of measurement is $(\bar{X}_n - \bar{Y}_n)/\sqrt{Var(\bar{X}_n - \bar{Y}_n)}$.

Now, we know,

$$Var(\bar{X}_n - \bar{Y}_n) = Var(\bar{X}_n) + Var(\bar{Y}_n) - 2Cov(\bar{X}_n, \bar{Y}_n) \tag{5.3}$$

If it is assumed that $\bar{X}_n$ and $\bar{Y}_n$ are uncorrelated, then,

$$Var(\bar{X}_n - \bar{Y}_n) = Var(\bar{X}_n) + Var(\bar{Y}_n) \tag{5.4}$$

This assumption might be conservative, as it is possible that $Cov(\bar{X}_n, \bar{Y}_n) > 0$. For example, since we shifted each window 0.2 *s* each time, we have a total of 501 overlapping windows for each game in $Session_6$ (recall that each game lasted approximately 120 *s*).

Now, if we consider $\bar{X}_n$ represents $Game_1$ and $\bar{Y}_n$ represents $Game_2$, then for $Session_6$, then,

$$\sqrt{Var(\bar{X}_n - \bar{Y}_n)} \leq \sqrt{\frac{(0.07)^2}{501} + \frac{(0.05)^2}{501}} = 0.0038.$$

Thus, the measurement of precision of this session is:

$$\frac{(\bar{X}_n - \bar{Y}_n)}{\sqrt{Var(\bar{X}_n - \bar{Y}_n)}} \approx \frac{(0.47 - 0.50)}{0.0038} = \frac{-0.03}{0.0038} = -7.8 s.d.s$$

This implies that it is highly unlikely that $\mu_X > \mu_Y$. We present the measurement of precision for each *Session* in Table 5.1.

From these values, one can see that the measures produced by our method for two games of a session are different at least by an order of magnitude of the standard deviation. Although two games of a session show a small difference in their synchronization index values, actually the differences are significant.

## 5.2 Validation of our method through comparison with an alternative ES method

Our method takes multiple task level events into account to measure the synchronization of the group. One may wonder if this approach is comparable or more accurate than singular event-based methods in the literature. Thus, we first validated our approach by comparing it to the method proposed by Varni et al. ([219]). As discussed in Section 4.1.3, this is a singular event-based synchronization detection method and represents a reasonable point of comparison.

### 5.2.1 Data collection

To measure group synchrony using the method by Varni et al. ([219]), we used the same data as described in Section 5.1.2. We considered the same six experimental sessions, consisting of two games in each session for the comparison. Here, we incorporated the skeletal data of the participants, which is aligned with the approach Varni et al. employ in their paper.

### 5.2.2 Method description and event detection

To employ the method by Varni et al. ([219]), one first measures the pair-wise synchronization index of two participants from a single type of event. From these indices, a connectivity graph is generated and then the group synchronization index is calculated.

In their work, Varni et al. ([219]) measured the class of events from participants' body motion features. As described in Varni and Camurri [218], these body motion features might include the contraction index, fluidity index, etc; however, in their recent work they used motion index (MI). The authors reported calculating MI by performing silhouette-based background subtraction; however, for our comparison study we extracted MI using upper-joint skeletal data, as this was a more robust measure given the overall background illumination of our dataset.

To calculate MI, we first projected the 3D skeletal coordinates to the 2D image plane. Then, from two consecutive frames, we calculated the distance each upper body joint moved for each participant during the game. If any joint position was poorly tracked due to occlusion, we discarded that joint movement from the calculation, using a comprable exclusion method as described in Section 5.1.3.

During the next available well-tracked position of that joint, we measured the distance moved from the previously well-tracked frame by that joint and divided the value by the number of consecutive poorly-tracked frames for the calculation. We took the local maxima of the sum of the distances of all the upper body joints' movements in a sliding window of $n$ frames as our event class.

Here, we used a window size of 3 ($n = 3$). We removed local maximas with a value less than 200 pixels as noise. To calculate the group synchronization indices, we used the same time window ($20s$), $\tau = 0.21s$ and $Q_{thresh} = 0.35$ as used with our method.

These events represent when a participant moves their upper body the most within a time window. It can approximate any abrupt body movements, or even pulsing

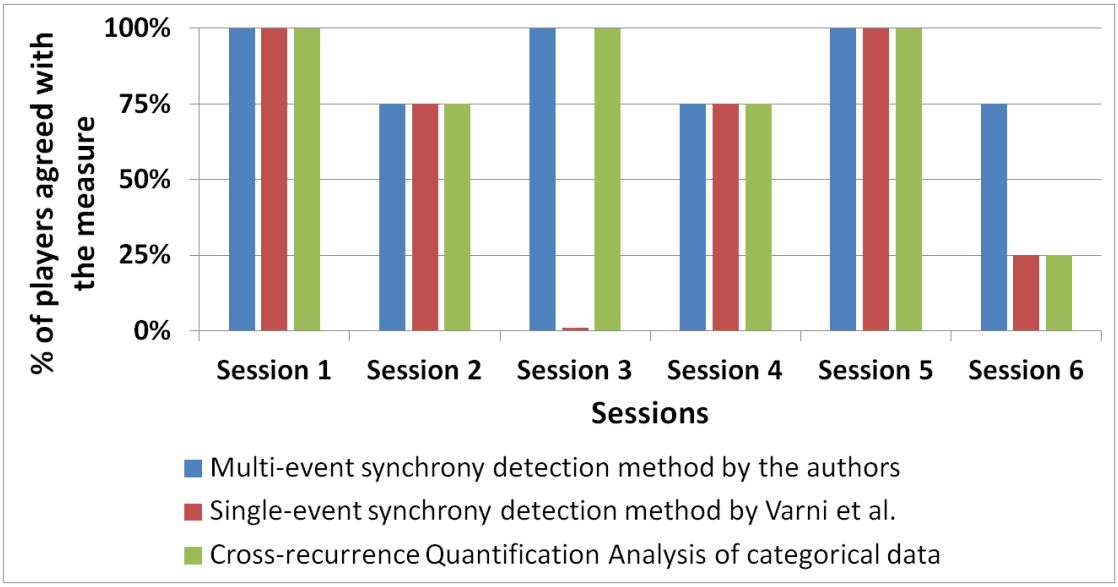


**Figure 5.5**: Agreement between players in each session and the synchrony measures (our measure, the measure by Varni et al. ([218]), and the CRQA measure). One can see that the majority of the participants (at least 75%) agreed with the measurement produced by our method in 100% cases (6 out of 6 sessions).

movements made in synchronous rhythm with the game. Thus, this class of events was similar to the class of events generated using the MI by Varni et al. ([219]) in their original experiment.

### 5.2.3 Results

Table 5.2 shows the group synchronization indices measured using the method proposed by Varni et al., as well as our method for all six experimental sessions. The table shows the values of the synchronization indices averaged over the duration of each game, as well as the standard deviation (s.d.) for the group synchronization indices for every game.

In Fig. 5.5, we present the percentage of players in each session for which the synchrony detection measures produced a match with the players' perception. For example, for $Session_3$, the group synchronization indices produced by our method are

0.44 and 0.48 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous; thus indicating 100% agreement. In contrast, the group synchronization indices produced by Varni el al.'s method are 0.70 for $Game_1$ and 0.67 for $Game_2$; indicating 0% agreement with players' perceptions.

### 5.2.4 Discussion

From the group synchronization indices presented in Table 5.2, one can see that the values measured using the Varni et al. method are higher in degree than the values produced by our method for all the games. These higher values might indicate an overestimation of the synchronization indices, as only a singular type of event is considered in the Varni et al. method. Also, as our method considers multiple types of task-level events, it may be more conservative in nature.

As one can see from the data presented in Fig. 5.5, the group synchronization indices produced by the method proposed by Varni et al. agreed with the perceptions of the majority of participants in only 66.67% of the sessions (4 out of 6 sessions), whereas our method agrees with participants' perceptions in 100% of the sessions (6 out of 6 sessions).

Moreover, the standard deviation values in Table 5.2 also suggest that our method was more accurate in assessing group synchrony stability than the method proposed by Varni et al. (As a reminder, a lower s.d. value reflects more stable synchrony, and a higher s.d. value reflects weaker synchrony). If the s.d. values are equal for both games (e.g., see $Session_2$), we consider them to be aligned with the perceptions of participants as a group. The s.d. values for the method by Varni et al. are aligned with participants as a group only in 50% of cases (3 out of 6 sessions), whereas our results are aligned in 83.33% of cases (5 out of 6 sessions).

## 5.3 Validation of our method through a comparison with CRQA

Dale et al. ([35]) used the cross-recurrence analysis method for quantifying the relationship between two categorical time series data through use of a contingency table. Recently, Coco et al. ([31]) released a package in *R* that implements CRQA and other methods. In this section, we use this package to perform a comparison between CRQA and our method.

### 5.3.1 Data collection

We used the same cup game data as described in Section 5.1.2. We considered the same six experimental sessions, consisting of two games per session.

However, the data contains instances where none of the synchronous events we were measuring occurred during a given moment in time. This "non-event" condition may have happened in between existing events, such as between clapping or tapping events. Our method is capable of supporting non-events, however, were we to use these data within CRQA, there is a chance these data will overestimate the recurrence profile of the time series. (i.e., both the pair-wise and group synchrony indices would be artificially inflated). To avoid this potential chance for overestimation, we performed a pre-processing step on each time series pair to remove any instances where "non-events" occurred both both players.

### 5.3.2 Description of the analysis

The *'crqa' R* package by [31] contains a function named *CTcrqa* which uses contingency tables (CT) to perform a cross-recurrence analysis on categorical data. First,

it finds all the categories from both time series. In our data, different categories are different event types. Then it calculates all the co-occurrences of different sets of events between those two time series to build a CT. After that, the recurrence profile is computed along the diagonal of the CT [31]. Different delays can be used to generate different recurrence profiles for the time series.

We used *CTcrqa* to compute the cross-recurrence profile for our data. Two delays were used to compute the recurrence profile. First, we used a delay of 0, which means the package computed the co-occurrences of the same event types for both time series. Then, we used a delay of 1, as we also wanted to count events as synchronous if they appeared consecutively in the time series.

For each time series pair, we first calculated the cross-recurrence of the pair for each delay. Then, we took the average of these values for the two delay patterns (0 and 1) as the measure of the pair-wise synchrony between these two time series. We used the same procedure for all of the pairs to calculate the pair-wise synchronization index.

After computing the pair-wise synchronization indices, we computed the individual and group synchronization indices for each game by following the method described in Section 4.1.3. First, we built group topology graph to calculate the individual synchronization indices. Then, from the connectivity value and individual synchronization indices, we computed the group synchronization index of that group. Based of the values produced by CRQA, we used $Q_{thres} = 0.1$ to generate the *group topology graph*. We did not use any sliding windows during these calculations.

### 5.3.3 Results

Table 5.2 also shows the group synchronization indices measured using CRQA for all six experimental sessions. In Fig. 5.5, we present the percentage of players in each session for which the synchrony detection measures produced a match with the players'

perception.

For example, for $Session_6$, the group synchronization indices produced by our method are 0.47 and 0.50 respectively for $Game_1$ and $Game_2$, and all players rated $Game_2$ as more synchronous, thus indicating 100% agreement. In contrast, the group synchronization indices produced by CRQA are 0.18 and 0.17 respectively for $Game_1$ and $Game_2$, thus indicating 0% agreement with players' perceptions.

### 5.3.4 Discussion

CRQA was slightly more accurate than Varni et al.'s method in assessing group synchrony, as it reached agreement with participants in 5 of 6 sessions. However, one can see in Table 5.1, there was a significant difference between the two games in $Session_6$ and players' perceptions (which we take as ground truth), suggesting that $Game_2$ was more synchronous than $Game_1$. Our method concurred with players on this assessment, suggesting it is more accurate than the other two methods.

One also may observe that the synchronization indices produced by the CRQA method are fairly low. This might happen as we used the relative ordering of the events during CRQA, instead of an equally sampled time series (which is what the CRQA measure assumes). Thus, there might be cases when the same events occurred in both time series slightly apart in the event order, but not as co-occurent or consecutive events. These events might be considered as non co-occurent events, which might be the cause of the CRQA synchronization index to be lower.

## 5.4 General discussion

In this chapter, we described three experimental validations of the method. First, we successfully applied it to a synchronous tabletop game using fixed sensors to detect

both individual and group synchronization indices. Our method closely matched players' own perceptions of the group synchrony across multiple games.

Then, using the same data, we compared our method to a single-task level method in the literature, and showed our approach was more accurate overall. As our method incorporates multiple task level events, it is more conservative than the comparison method. In general, the synchronization indices measured using the comparison method were larger in value than ours, and were less likely to match players' perceptions of synchrony.

We also compared our method to a categorical approach from the literature, cross-recurrence quantification analysis (CRQA), and again demonstrated our approach yields more accurate results. Furthermore, unlike CRQA, our method is capable of dealing with "non-events", i.e., events that occur between synchronous events, and is thus also provides a more conservative estimation of synchrony than CRQA. Indeed, failing to conduct a pre-processing step on the types of data we work with (psychomotor synchrony), CRQA is likely to overestimate both the pair-wise and group synchronization.

Our method presents several advantages. First, it is simple, fast, and suitable for online implementation. In our lab, we have recently implemented this method to work on an autonomous mobile robot to measure group synchrony, and move in synchrony with others in real time [166]. We plan to release this software as open source in the near future.

Second, new types of events can be easily added without requiring any changes to the algorithm. This enables great flexibility, should a researcher wish to explore increasing the granularity of synchronous events, or want to incorporate new types of social and/or affective behaviors.

Third, the method is robust, and works successfully with data from both fixed and mobile sensors. It also works using data from unimodal or multimodal sensors. This

robustness is particularly advantageous if a researcher is fusing synchronous data from sensors with varying frame rates.

In addition to group synchrony detection, our method can also be used to measure each participant's individual synchrony. Some group members may be more likely to be "team players", tending to synchronize more readily than others, and it could be useful to detect their role in how group synchrony emerges. This may particularly prove useful in the field of psychiatry, where researchers are interested in assessing how individuals with schizophrenia physically interact in groups of matched controls, and how they experience non-verbal social exclusion [118, 119]. It also is useful to researchers in psychology, who often seek methods to study rapport building, social encounter smoothing, and cooperation efficiency, all of which depend on how well one can synchronize to others [38].

Our method can be helpful for other researchers in the affective computing community in several ways. First, our method can enable the next generation of human-machine systems to estimate the affective behavior of a group as a whole, as well as individual group members, by assessing how well-entrained they are to one another. This has broad applications across the field - such as in dominance detection in groups [183]), the affective behavior of crowds [132], assessing the emergence of group roles [167].

Second, it is robust to include many different types of events from multimodal data sources; few methods exist in the field to allow for this. The field in general has been trending toward multimodal affect modeling and social scene understanding (c.f. [21, 39, 221]; this work provides both a theoretical and practical contribution in this area.

Finally, our method could be used to detect engagement and skill acquisition in affective learning contexts. Many researchers in this community seek to ensure ecological validity in their work [2]; for most students the naturalistic learning context involves relationships with peers [57]. The ability to model the behavior of a student within the

context of group learning could be useful to those who work in this space.

## 5.5 Chapter summary

In this chapter, we validated our method in three ways. Through these validations, we tried to address our second research question. The results suggest that our method can successfully measure group synchrony, and is more accurate compared to other methods from the literature. In next chapter, we will validate our method through a human-robot teamwork scenario. We also will explore whether such a measure can be used to estimate asynchronous behavior.

## 5.6 Acknowledgements

I thank James Delaney, Giovanna Varni, Moreno Coco, Olivia Choudhury, Michael Gonzales, Cory Hayes, and Maria O'Connor for their valuable suggestions during the algorithm development phase, the study design, and the experimental setup.

This chapter contains material from "Assessing group synchrony during a rhythmic social activity: A systemic approach," by T. Iqbal and L. D. Riek, which appears in Proceedings of the Conference of the International Society for Gesture Studies (ISGS), 2014 [80], "A Method for Automatic Detection of Psychomotor Entrainment," by T. Iqbal and L. D. Riek, which appears in IEEE Transaction on Affective Computing, 2015 [82], and "Detecting and Synthesizing Synchronous Joint Action in Human-Robot Teams," by T. Iqbal and L. D. Riek, which appears in Proceedings of the International Conference on Multimodal Interaction, 2015 [83]. I was the primary investigator and author of these papers.

# Chapter 6

# Validation of the Method with a Human-Robot Team

In order for robots to competently and contingently collaborate with humans, they need to be able to solve the challenge of sensing high-level human activities occurring in their surroundings. This is particularly important when both the robots and people are moving [27, 176].

Researchers from the activity recognition and robotics communities generally combine *motor primitives* to understand a wide range of complex actions [92, 45, 144, 196, 197]. While this work is useful for some perceptual situations, from a human-robot teamwork perspective, it may not tell robots much information about the *context* of how humans are interacting within the environment and with one another. This makes it challenging for robots to respond appropriately, and even more so when both the robots and co-present humans are moving [176].

If robots are to obtain this capability, however, they must first be able to accurately sense synchronous action occurring around them, ideally while in motion. To enable this, we present a method to automatically measure synchronous joint action between people

as observed from mobile robots.

In this chapter, we aim to address our third research question described in Chapter 4, by validating our method against a rhythmic group activity (marching), as observed by two autonomous mobile robots (see Section 6.1). In Section 6.2 we present the results from this validation, and in Section 8.5 discuss their implications for the research community.

## 6.1 Method validation through a human-robot interaction scenario

In previous chapters, we validated our method by observing groups of people performing a synchronized psychomotor activity ("the cup game") using fixed sensors. Here, we wanted to extend this work to explore how well our method worked with mobile sensors (robots) and mobile people. We were also interested in exploring asynchronous conditions. Thus, we analyzed the synchrony of people marching as followed by two mobile robots.

Marching is a group activity that is both dynamic as well as rhythmic in nature [126]. We conducted a set of controlled experiments where we had two individuals perform four different sets of marching actions (see Fig. 6.2). In each experiment, a mobile robot followed each performer to detect different events while the person was marching (see Fig. 6.1).

We used two Turtlebot robots as mobile platforms in our experiment, and used their attached Kinect sensors for capturing data. The Turtlebot is an open-hardware and software platform comprised of an iRobot Create platform, a Microsoft Kinect, and an ASUS laptop running the Groovy version of the Robot Operating System (ROS) on Ubuntu Linux [184, 175].

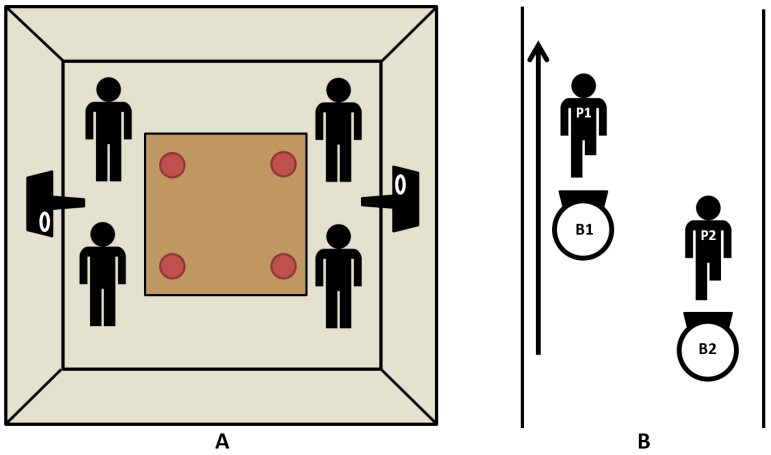


**Figure 6.1**: A) First validation measured people performing synchronous joint action in a static setup [82]. B) Current validation method has autonomous mobile robots following humans moving synchronously. P1 and P2 are the performers, and B1 and B2 are the robots.

Prior to our experiments, we adjusted the Kinect sensors on our Turtlebots so that they could track the performer's feet when they stepped and raised each foot. We used the TurtleBot's "Follower" program so that our robots could follow people autonomously while recording data, and stored the data in the ROS bag format.

We recruited two naive performers by word-of-mouth for our validation experiments. We gave both performers instructions for marching, and conducted one practice session before each of our four marching scenarios. We recorded two sessions for each of our four scenarios for analysis to account for noise.

In each scenario, we instructed each person to perform a "high-march", which constitutes picking up their knees and feet to an exaggerated degree. The first performer led the march, and was situated on the left side of the hallway. We situated the second performer behind and to the right of the first marcher as they moved down the hallway so that they could adjust their actions based on those of the first performer (see Fig. 6.1). All scenarios lasted approximately 35 seconds, and was timed using a stopwatch. We gave the first performer an mp3 player with a set of noise-cancelling headphones playing

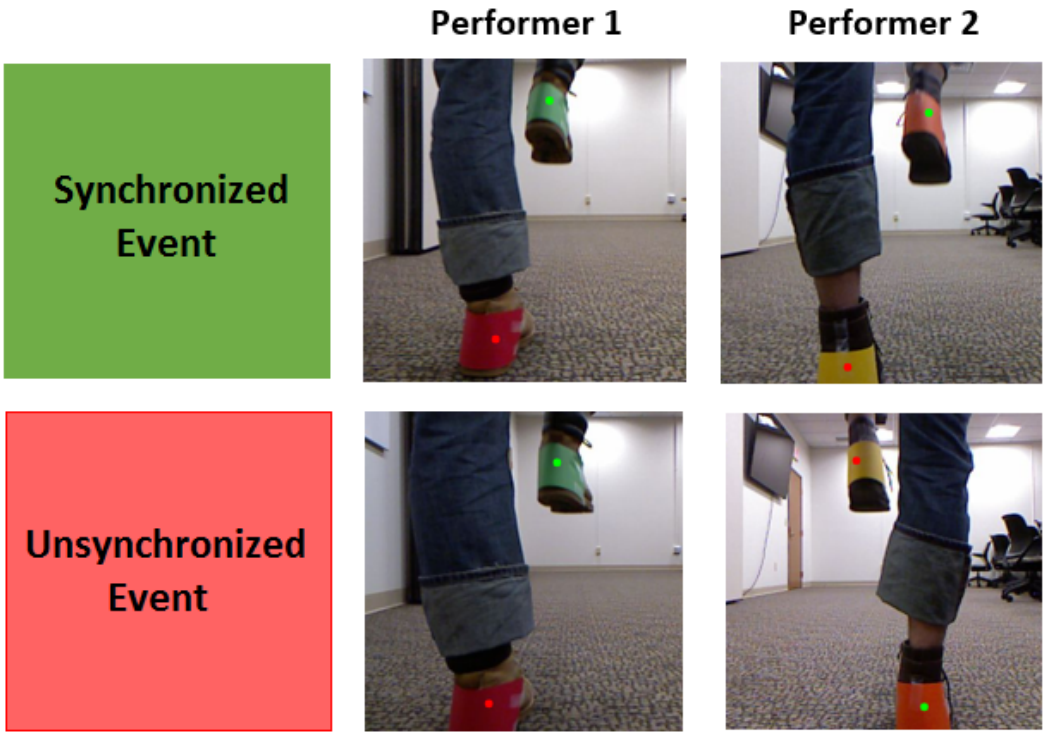


**Figure 6.2**: A comparison of synchronous and asynchronous events captured by the mobile robots. Dots represent the tracked positions of the feet.

John Philip Sousa's "Stars and Stripes Forever" march to keep tempo and prevent any noise-related distractions. Since the second performer follows the first performer from behind, the marching pace or pattern of the second performer did not affect the marching pace or pattern of the first performer.

In the first experimental scenario, the performers marched down a hallway in a synchronized manner. During this scenario, the second marcher followed the steps of the first marcher. In the second scenario, the second marcher followed the steps of the first marcher, but did so in the opposite order. For example, when the first marcher raised their left foot, the second marcher raised their right.

For our third scenario, both performers started out marching in a synchronous fashion, became asynchronous after 12 seconds, then synchronous again after 24 seconds. The fourth scenario reversed the actions of the third scenario, where both marchers began

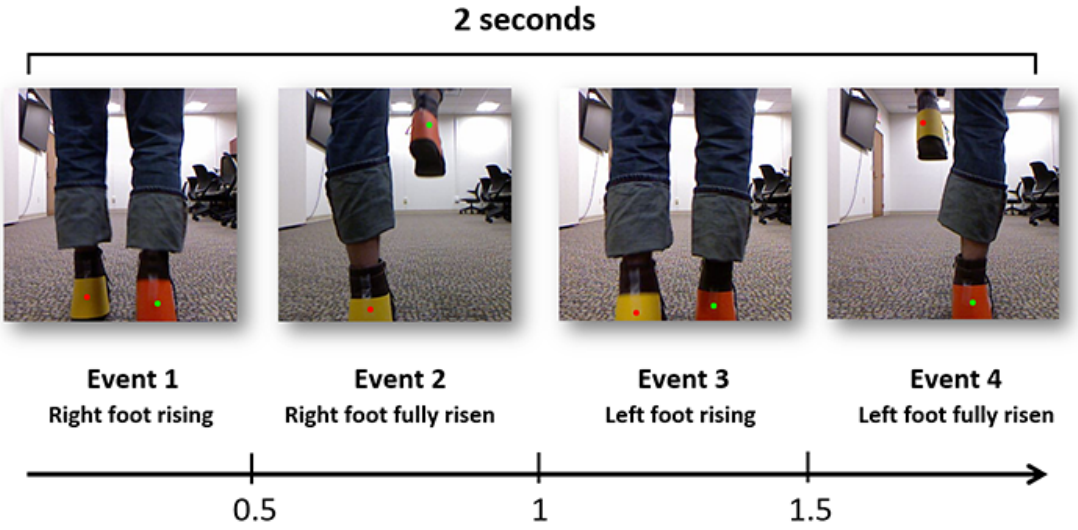


**Figure 6.3**: A decomposition of events one through four that are used in event synchrony measurement. Overall, all events occur in two seconds.

unsynchronized, became synchronized after 12 seconds, and unsynchronized after 24 seconds. We verbally instructed the second performer when to switch their steps to after these time periods by saying "switch". Fig. 6.2 shows the synchronous and asynchronous marching patterns.

### 6.1.1 Data collection

Before recording data, both systems were synchronized with an Ubuntu time-server to ensure that both systems kept accurate timing. The videos of each performer were time-synchronized, and analyzed using our model to determine the group synchrony based on the occurrence of events.

To measure the overall group synchrony, we had to first determine the different task-level events of this group activity (i.e., a leg raise, or a leg leaving the ground) from the recorded videos. For this experimental setup, we defined two types of events to measure overall group synchrony. The first type of event was when a person begins to

raise his/her a leg from the ground. The second type of event was when a leg reaches its maximum height. As a result, a total of four types of events occur when a person is marching (one of the aforementioned events for each leg). See Fig. 6.3 for an example of these events.

To track each performer's feet, we used a standard blob tracking technique. We used the ROS cmvision package to track color blobs in RGB images. We attached four unique small squares of colored paper (orange, yellow, green, and red) to the performers' left and right feet, while each robot followed behind each performer at a distance of two feet to track each performer's actions.

Fig. 6.5-A shows the expected synchrony for these four scenarios. We expected to see a high value for a synchronization index for the entire duration of a session for Scenario 1, and a value of zero for Scenario 2. For Scenario 3, we expected to see our measured synchronization index decrease beginning around seven seconds to a value of zero at 12 seconds, and increase again at about 20 seconds. For Scenario 4, we expected similar results, however in reverse order.

### 6.1.2 Data analysis

From all detected blob positions, we first discarded very small blobs as noise. The remaining blob positions were considered candidates for a foot position. From these candidate blob positions, we generated an undirected graph by treating each blob center as a vertex of that graph. Two vertices were connected in this graph with an edge if they were closer to each other than a threshold distance.

From the resultant graph, we calculated the connected components. We then clustered all connected components together, and measured the total area of all blobs in each cluster. The cluster of blobs with the largest area was defined as the location of the foot in the RGB image. The center of that cluster, calculated by taking the mean position

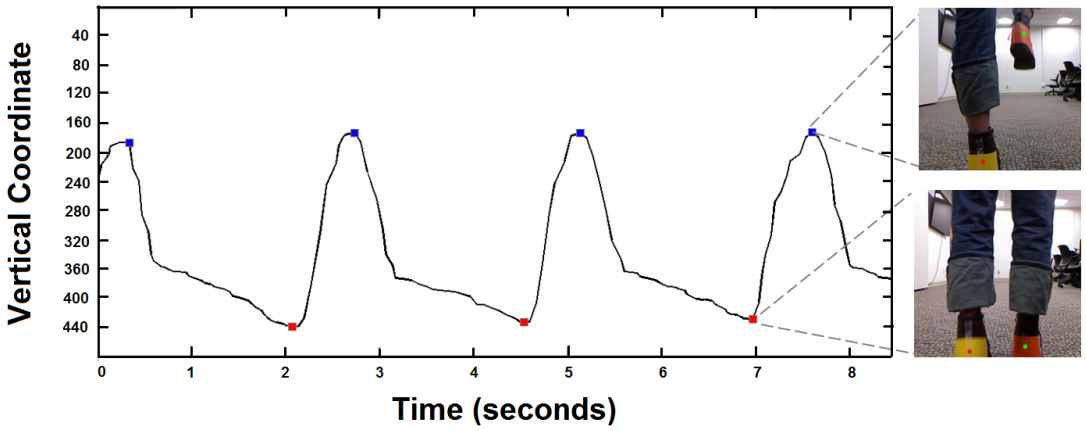


**Figure 6.4**: This figure shows the detection of each event for one foot to show when a person's foot is leaving the ground, and when their foot reaches its maximum height.

of each blob of that cluster weighted by each blob's area, is considered the foot position.

We then detected the high-level events in this rhythmic activity from the movements of each performer's feet. While marching, the foot position begins to move in an upward direction when the foot leaves the ground and eventually reaches its maxima before descending back downward [128, 72]. From the recorded video from the mobile robots, we found that this phenomenon also holds.

In our recorded video, the position of each foot changes significantly along the vertical axis of the RGB image plane while marching. Since the robot is also moving, we must extract the movements of the feet without regard to the robot's movement. Due to the ego-motion of the mobile robot and the attached camera, the position of the feet changed in every frame along both axes, although the feet were stationary in the real world. However, in the case of marching (raising the feet from the ground up), the change of the position of the feet along the vertical direction is significantly larger than the changes in position in the RGB image caused by the ego-motion.

To detect high-level events, we calculated the local extrema (maxima and minima) of the positions of the feet along the vertical axis. While calculating the local extrema,

we use an additional condition that discarded any extrema that occurred within a specific period of time. This additional condition helps remove noise due to poor tracking or drastic movement caused by the robot's motion.

Due to the robot's ego-motion, the changes of the feet positions along the vertical axis were less than when they performed marching steps. An example of this effect can be seen between 3.2 and 4.6 seconds in Fig. 6.4. Our measure also accounts for this effect.

Our local minima and maxima are used to give us the time when a foot starts to leave the ground and when it reaches its maximum height. From these values, we are able to define that an event occurred during that time. This measure is independent to the ego-motion of the robot in our setup, as well as the height and pacing of each performer's movement. Fig. 6.4 describes this event detection model.

From the recorded video, we detected the events for both performers. We then measured the overall group synchrony using the model described in Chapter 4.

## 6.2 Results

To sufficiently measure synchrony, we found a five second window to be ideal. This is due to the fact that each performer needed about one second to complete one step, and around two seconds for all four events to occur (one for each foot leaving the ground, and one for each foot reaching its maximum height, as shown in Fig. 6.3). We used $\tau = 0.21s$ and a sliding window of $5s$ for the calculation.

Each march session lasted about 35 seconds. Fig. 6.5 shows the synchronization index of each session over time. Here, four sessions are presented in four different graphs. As we used a sliding window size of five seconds, the synchronization index value for the zeroth second actually represents the synchronization index value on the window from

zero to five seconds.

In Fig. 6.5, one can see that the synchronization index of the group is approximately 0.7 for the first experimental scenario, where the second performer was instructed to synchronously follow the steps of the first performer. In the case that the performers are asynchronous in their movements, then the synchronization index will be zero. On the other hand, if movement is very synchronous, then their synchronization index will be close to one.

The synchronization index of the second scenario is also presented in Fig. 6.5, which shows that the synchronization index is zero across the entire session. This indicates that during the time that our performers were marching, no event occurred in-sync over the entire session.

For the third experimental scenario, one can see that for the start of the session, the synchronization index was approximately 0.7 for the first few seconds, as shown in Fig. 6.5. This is due to the fact that we instructed performers to begin this session by marching in a synchronous manner. After 12 seconds, we verbally instructed the second performer to switch their steps and become asynchronous with the first performer, causing the synchronization index to decrease. After 24 seconds, we again verbally instructed the second performer to change their synchrony with the first performer using the term “switch”. This caused an increase in the synchronization index.

From Fig. 6.5, we can also see that the synchronization index became zero around the 12th second, which shows that the synchronization index for the five second window starting at the 12th second was zero. As we are using a sliding window of five seconds, the synchronization index began to decrease after approximately seven seconds. Similarly, when the second performer became synchronous with the first performer at the 24 second mark, we see a increasing synchronization index value starting at the 19th second mark.

For the fourth experimental scenario, one can see that the synchronization index

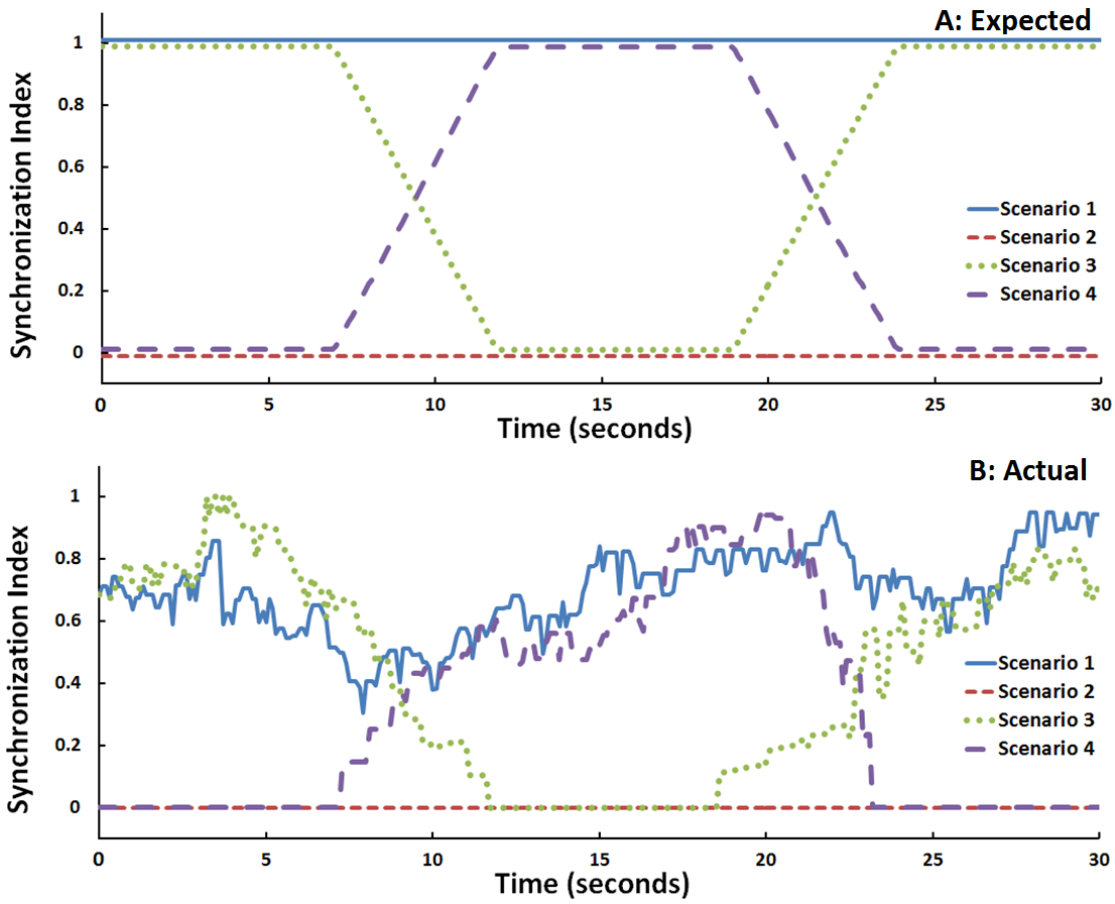


**Figure 6.5**: A) The expected synchronization indices of our experimental scenarios. B) Actual synchronization indices of our experimental scenarios.

was zero at the start of the session. This is because the performers began marching asynchronously until approximately the 12 second mark, where they became synchronous. After about the 24th second, marchers became asynchronous again. From the graph, we also see that the synchronization index is higher in the case of synchronous movements, which started to increase around the seventh second in the graph due to the sliding window algorithm. The synchronization index starts to decrease after the 20th second mark when movements are asynchronous after the 24th second, thereby dropping the synchronization index to zero.

We present the mean value of the synchronization indices of the sliding windows

**Table 6.1**: Mean Synchronization Indices During Each Interval of Marching

| Time | Mean Synchronization Index | | | |
|---|---|---|---|---|
| | **Scenario 1** | **Scenario 2** | **Scenario 3** | **Scenario 4** |
| **0-12s** | 0.67 | 0 | 0.8 | 0 |
| **12-24s** | 0.68 | 0 | 0.18 | 0.55 |
| **24-35s** | 0.79 | 0 | 0.68 | 0 |

in Table 6.1. In the first row, we present the average value of the synchronization indices of the sliding windows starting from zero to seven seconds. These sliding windows originally cover all of the events that occur between zero to 12 seconds. The second row shows the values of the sliding windows from the seventh to the 19th, and the last row presents from the 19th to the 30th second. The last two intervals represent the events occurring from the 12th to 24th second, and 24th to 35th second in real-time, respectively. We use these intervals because we instructed our second marcher to switch the marching pattern at the 12th and the 24th second during the third and fourth scenarios.

From Table 6.1 one can see that the mean value of synchronization indices does not change much for Scenario 1 and 2, since the marching pattern did not change during these cases. For Scenario 1, the mean synchronization indices are high for all three time intervals. On the other hand, for all three intervals, we see the values are zero for Scenario 2. We see changes of the synchronization index value for Scenario 3 and 4 in different intervals. For Scenario 3, we observe lower values for the middle interval when the performers were asynchronous, and higher values for first and last interval when the performers were synchronous. We observe the opposite pattern for Scenario 4, with a higher value in the middle interval when the performers were synchronous, and lower in the first and the last interval where the performers were asynchronous.

## 6.3 Discussion

As shown in Fig. 6.5, the synchronization indices for the four scenarios matched our expectations. The first scenario (synchronized) showed a high synchronization index (0.67) for the entire duration of the session. The second scenario (unsynchronized) similarly matched our expectations, with a value of zero for the entirety of the session. The third scenario (mixed, with a synchronized start) showed high values for time-frames that the second performer was instructed to be synchronized, and a value of zero for the time frame they were instructed to be unsynchronized. Scenario four (mixed, with an unsynchronized start) also matched these expectations.

Our results suggest that our method is effective in capturing and processing synchronized joint action occurring with both robots and people in motion. The results also suggest that our method can successfully measure asynchronous conditions (scenario 2, 3, and 4).

This work is encouraging for future work in understanding team dynamics in real-time for robotic systems. This work will also help in the design of robot behavior that generates synchronous joint action. Considering motion may distort sensing, our results show that our method was capable of detecting synchronized events and measuring synchronous joint action between two individuals in motion, independent of the height and pacing of steps.

While we expected a higher synchronization index value for Scenario 1, this could be due to a variety of reasons. Lighting and illumination variation may have an effect on blob tracking, especially since the robot and person are constantly in motion. Such changes may affect color calibration and tracking, causing the algorithm to miss a correct event or detect an incorrect event.

## 6.4 Chapter summary

In this chapter, we validated our method by applying it to a human-robot synchronous marching task. Through this validation study, we explored whether our synchronization detection measure can be used to estimate asynchronous behavior, The results suggest that our method can successfully estimate asynchronous behavior in a team, even when all the group members are in motion. In the next chapter, we will extend our method to introduce an anticipation method for robot control, which will take team dynamics into account.

## 6.5 Acknowledgements

I want to thank Michael J. Gonzales for helping me designing and performing the experiments, and during the writing phase. I also want to thank Olivia Choudhury, Mark Costanzo, Maryam Moosaei, and Aaron Steele for helping me during the experiments.

This chapter contains material from "Detecting and Synthesizing Synchronous Joint Action in Human-Robot Teams," by T. Iqbal and L. D. Riek, which appears in Proceedings of the International Conference on Multimodal Interaction, 2015 [83], "Role distribution in synchronous human-robot joint action," by T. Iqbal and L. D. Riek, which appears in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, Towards a Framework for Joint Action Workshop, 2014 [81], "Mobile robots and marching humans: Measuring synchronous joint action while in motion," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the AAAI Fall Symposium on AI-HRI, 2014 [75], and "Joint action perception to enable fluent human-robot teamwork," by T. Iqbal, M. J. Gonzales, and L. D. Riek, which appears in Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 2015 [76]. I was the primary investigator and author of these

papers.

# Chapter 7

# An anticipation method for robot control

In the previous chapters, I described a new method that enables a robot to understand team dynamics. Based on its understanding, if a robot can make predictions about the future actions of its human counterparts, then the robot will be able to interact more fluently within a group [65].

Recent work in robotics has focused on developing predictive methods for improving the fluency of joint interaction between a robot and one or more humans. For example, Hoffman et al. ([64]) proposed an adaptive action selection mechanism for a robot, which could make anticipatory decisions based on confidence of their validity and their relative risks. Additionally, Pérez-D'Arpino et al. ([157]) proposed a data-driven approach to synthesize anticipatory knowledge of human motion, which they used to predict target objects during reaching motions. Unhelkar et al. ([214]) extended this concept for a human-robot co-navigation task. This model used "human turn signals" during walking as anticipatory indicators, in order to predict human motion trajectories.

While this work will improve the ability of robots to have fluent interactions

within HSEs, most of these methods are best-suited for dyadic interaction and dexterous manipulation contexts. In contrast, we seek to explore methods for robots that will work robustly in groups, and also for tasks involving gross motion with mobile robots.

In the previous chapters, we explored the problem of automatically modelling and detecting synchronous joint action (SJA) in human teams, using both fixed and mobile sensors. We introduced and validated a new, nonlinear dynamical method which performed more accurately and robustly than existing methods.

Now, we explore how a robot can use these models to synthesize SJA in order to coordinate its movements with a human team. The main contribution of this work is the introduction of a new anticipation method for robot motion to predict events and the time that they occur during a human-robot interaction scenario.

## 7.1 Event synchronization methods

For this work, we created two anticipation methods for robot motion. The first method, synchronization-index based anticipation (SIA), is inspired by our prior SJA detection work described in Chapter 4. It calculates the synchronicity of the group in real-time, determines who the most synchronous person is, and uses that information to move the robot.

The second method, event cluster based anticipation (ECA), we created to establish a reasonable comparison anticipation method for SIA that does not rely on team dynamics. ECA is a straightforward method that involves averaging the times participants moved during a previous iteration of the dance. Figure 7.1 gives a visual comparison of how the two methods work in practice, and they are described textually below.

In order to explain the application of the methods, we considered a synchronous dance scenario In this scenario, a team of humans and a robot coordinate their motion in

real-time. The dance is iterative, and performed cyclically in a counter-clockwise manner. There are four *iterations* in a dance session. In this chapter, we describe the methods in detail, and explain how we can use these methods in the testbed scenario.

## 7.2 Synchronization index based anticipation (SIA)

The Synchronization Index Based Anticipation (SIA) method takes a team's internal dynamics into account when generating robot movements. The main idea is that for a given iteration, the participant who moves the most synchronously with the other members of the group is a good model for the robot to follow in order to be well-coordinated with the team. Furthermore, the method will adjust its identification of the most synchronous person after each iteration. Figure 7.1-B explains this method visually.

Thus, to generate future actions for the robot, at the beginning of each iteration we measured the most synchronous person of the group using our method described in Chapter 4. We will briefly describe the method in Sections 7.2.1, and then discuss in Section 7.2.2 how we used it to assess the most synchronous person to inform how the robot should move.

### 7.2.1 Measuring the individual synchronization index

As described in Chapter 4, we can express the task-level events associated with each dancer as a time series. Suppose $x_n$ and $y_n$ are two time series, where $n = 1 \ldots N$. Here, each time series has $N$ samples. Suppose, $m_x$ and $m_y$ are the number of events that occur in time series $x$ and $y$ respectively, and $E$ is the set of all events [82].

Now, as described in Equation 4.5, we first calculate $Q_\tau(e)$, which represents the synchronization of events in two time series, where we only consider a single type of event $e$ in both time series.

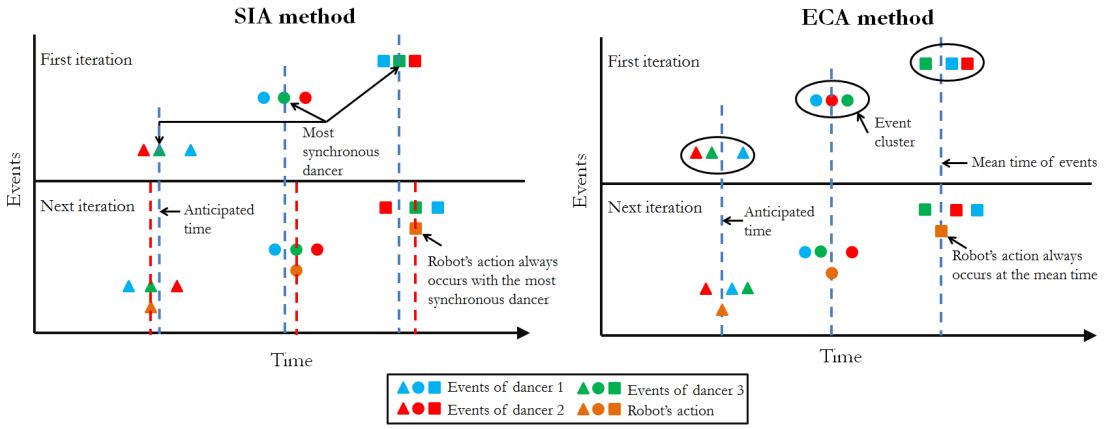


**Figure 7.1**: A visualization of the two anticipation methods. Left: Synchronization Index Based Anticipation (SIA), Right: Event Cluster Based Anticipation (ECA). The upper portion of the graph represents one iteration of the dance session, and the lower portion represents the next iteration of the same session.

The value of $Q_\tau(e)$ should be between 0 and 1 ($0 \leq Q_\tau(e) \leq 1$), as we normalize it by the number of events that appear in both time series. $Q_\tau(e) = 1$ shows that all the events of both time series are fully synchronized, and appeared within a time lag $\pm\tau$ on both time series. On the other hand, $Q_\tau(e) = 0$ shows us that the events are asynchronous [82].

Now, we measure the synchronization of events in two time series for multiple types of events, as described in Equation 4.6. So, the overall synchronization of events in time series $x$ and $y$ of that pair is:

$$\forall e_i \in E(n) : Q_\tau^{xy} = \frac{\sum [Q_\tau(e_i) \times [m_x(e_i) + m_y(e_i)]]}{\sum [m_x(e_i) + m_y(e_i)]} \tag{7.1}$$

If all events are synchronous in both time series, then the value of $Q_\tau^{xy}$ will be 1. If they are asynchronous, the value of $Q_\tau^{xy}$ will be 0 [82].

We calculated the pairwise synchronization index for each pair. Suppose we have $H$ number of time series. The time series data are represented as $s_1, s_2, \ldots, s_H$. First, we calculate the pairwise event synchronization index for each pair. So, we have the value

of $Q_\tau^{s_1 s_2}, Q_\tau^{s_1 s_3}, \ldots, Q_\tau^{s_{(H-1)} s_H}$.

We modified our process slightly from that described in Chapter 4. After calculating the pairwise synchronization index, we built a directed weighted graph from these indices, where each time series is represented by a vertex. However, in Chapter 4, after calculating the pairwise synchronization index, an undirected weighted graph was built. In a fully connected situation, the directed and the undirected graphs represent the same connectivity.

So, if the time series are $s_1, s_2, \ldots, s_H$, then there is a vertex in the graph which will correspond to a time series. We connect a pair of vertices with a weighted edge, based on their synchronization index value. In this case, there will be an incoming and an outgoing edge for each pair of vertices. We will refer to this graph as the *group topology graph (GTG)*, which is the same as the graph described in Chapter 4.

The individual synchronization index ($I_\tau(s_i)$) depends on both the group composition and the size of the group. We assumed that during this dance performance, each human participant may have some direct or indirect influences on the other human participants of the group. $I_\tau(s_i)$ for a participant is measured as the average of the weight of the outgoing edges to the corresponding vertex in the topology graph. So, the $I_\tau(s_i)$ of series $s_i$ is:

$$I_\tau(s_i) = \frac{\sum_{j=1,\ldots,H,\ j\neq i} Q_\tau^{s_i s_j} \times f(s_i, s_j)}{\sum_{j=1,\ldots,H,\ j\neq i} f(s_i, s_j)} \tag{7.2}$$

Where,

$$f(s_i, s_j) = \begin{cases} 1 & \text{iff } edge(s_i, s_j) \in GTG \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

### 7.2.2 Determining the most synchronous dancer and anticipating their next movement

The person with the highest individual synchronization index during an iteration is considered the most synchronous person of the group. This is because a high individual synchronization index indicates close synchronization with the other group members. Thus, let this person be *MSP*.

Suppose, during $itr_i$, we determine $MSP(itr_i)$ as the most synchronous dancer of the group. Now, assuming that a similar timing pattern of events will occur during the next iteration ($itr_{(i+1)}$), if the robot follows the events of the $MSP(itr_i)$, then the group will become more synchronous.

We can describe this concept mathematically. To reach a synchronous state, all events must occur very closely in time, i.e., within a time lag $\pm\tau$. Thus, we want to minimize the difference between event timings for each pair of agents. Now, if $\Delta t_{ij}$ represents the time difference of one event between agent $i$ and $j$, then our goal is:

$$\forall i, j \in H : minimize\,(\sum \Delta t_{ij}) \tag{7.4}$$

Now for our scenario, as shown in Figure 7.2, suppose *Dancer* 2 was the most synchronous person during one iteration ($itr_i$) of the dance session, i.e., $MSP(itr_i)$ was *Dancer* 2. Now, during $itr_{(i+1)}$, a similar timing pattern holds, and the timing of one particular event of the three dancers and the robot are $t_1$, $t_2$, $t_3$, and $t_R$ respectively. To reach a synchronous state, the following is required:

$$\begin{aligned} minimize\,(\sum \Delta t_{12} + \Delta t_{23} + \Delta t_{1R} \\ + \Delta t_{R3} + \Delta t_{13} + \Delta t_{R2}) \end{aligned} \tag{7.5}$$

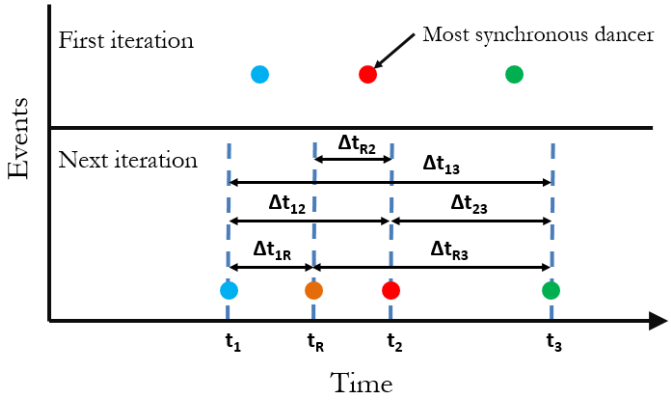


**Figure 7.2**: Example timings from a single type of event during two consecutive iterations.

As *Dancer* 2 is the *MSP*, from Fig. 7.2, one can see $\Delta t_{12} + \Delta t_{23} = \Delta t_{13}$, and $\Delta t_{1R} + \Delta t_{R3} = \Delta t_{13}$. Thus, Eq. 7.5 becomes:

$$minimize\,(\sum \Delta t_{13} + \Delta t_{13} + \Delta t_{13} + \Delta t_{R2}) \tag{7.6}$$

As only the term $\Delta t_{R2}$ depends on the robot's movement in Equation 7.6, by minimizing $\Delta t_{R2}$ we can minimize the equation. Thus, if the robot and the *Dancer* 2 (in this case, the *MSP*) perform the same event at the same time, then $\Delta t_{R2}$ will become 0, which will minimize Equation 7.6. This implies that if the robot can perform the events close to the timings of the most synchronous person, then the whole group will reach a more synchronous state.

Thus, for a given iteration, $itr \in \forall\, iterations$, the robot will determine $MSP(itr_i)$. As we know the timing of the events during the previous iteration of the dance $itr_{(i-1)}$, our anticipation method assumes that the similar event will happen more or less at the same time during this iteration $itr_i$. Therefore, when it is close to the timing of events of $MSP(itr_{(i-1)})$ during $itr_i$, and the robot receives any indication to the start of an event (we will describe this step in detail in Chapter 8) associated with $MSP(itr_{(i-1)})$, then the robot will anticipate those events as the indicator of the start of an event. The robot will

then perform the appropriate movement.

## 7.3 Event cluster-based anticipation method (ECA)

We created the Event Cluster-Based Anticipation Method (ECA) method to establish a reasonable comparison anticipation method for SIA that does not rely on team dynamics. ECA is theoretically simple, but powerful in nature. During a rhythmic and iterative task scenario, the timings of the events for one iteration are similar to events that happened in the previous iteration. Thus, we averaged the events timing during one iteration to predict the timing of those same events for the next iteration. Figure 7.1-A depicts this method visually.

First, for one iteration, we present all the events associated with the dancers by a time series. Then, we cluster all the similar types of events together that occur within a time threshold, ε. For example, for a single event $e$, we calculate the timing of the event performed by three group members, i.e., $t(dancer_1(itr_i),e)$, $t(dancer_2(itr_i),e)$, $t(dancer_3(itr_i),e)$. Here, $t$ represents the timing of an event, and $itr_i$ represents the iteration $i$.

After that for each cluster of similar events, we calculate the average time of all the events and use that time as the timing of the event for the next iteration. These events and the times are the predicted events and timing for the robot during the next iteration of this scenario. Thus, $t(robot(itr_{(i+1)}),e) = (t(dancer_1(itr_i),e) + t(dancer_2(itr_i),e) + t(dancer_3(itr_i),e))/3$. Based on all the events for next iteration, the robot generates appropriate actions.

## 7.4 Chapter summary

In this chapter, I discussed a new anticipation method (SIA) for a robot to predict events, and their timings, during a rhythmic, iterative human-robot interaction scenario. This method relies on team coordination dynamics for event anticipation. I also introduced a comparison method (ECA), which does not rely on team dynamics. In the next chapter, we will employ these methods on a human-robot synchronous dancing scenario, and will investigate the effects of these methods on the overall group coordination.

## 7.5 Acknowledgements

I want to thank Samantha Rack for helping me designing and implementing these methods during the algorithm development phase.

This chapter contains material from "Will you join the dance? Toward synchronous joint action in human robot teams," by T. Iqbal., S. Rack, and L. D. Riek, which appears in Proceedings of the Conference of Joint Action Meeting, 2015 [78], and "Movement coordination in human-robot teams: A dynamical systems approach," by T. Iqbal., S. Rack, and L. D. Riek, which appears in IEEE Transactions on Robotics, 2016 [79]. I was the primary investigator and author of these papers.

# Chapter 8

# Movement coordination in human-robot teams

In the previous chapter, I presented a new anticipation method (SIA) for a robot to predict the events and their timings during a rhythmic and iterative human-robot interaction scenario. This method relies on the team coordination dynamics for event anticipation. I also introduced another event anticipation method (ECA) which does not rely on team dynamics.

Now, in this chapter, we validate the SIA method on a task where an autonomous mobile robot observes a team of human dancers, and successfully and contingently coordinates its movements to "join the dance". We compare the results of the SIA method to move the robot with the ECA method.

I describe the system architecture and the experimental testbed in detail in Section 8.1. I then describe the experimental procedure, and the result analysis in Section 8.2 and 8.4.

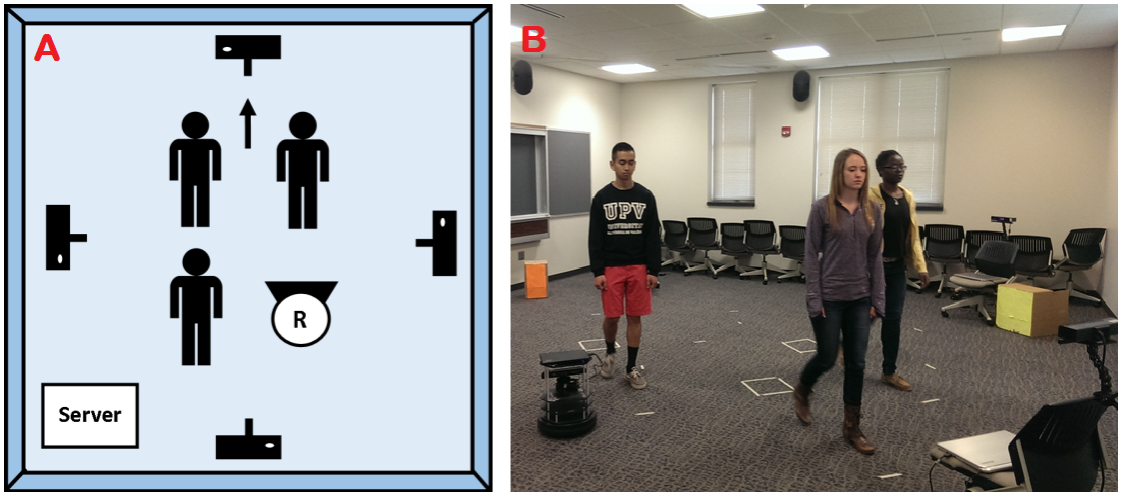


**Figure 8.1**: A) Data acquisition setup. B) Three participants are dancing along with a Turtlebot robot.

## 8.1 System architecture and experimental testbed

In order to explore how a robot can use human team dynamics to synthesize synchronous joint (SJA) action with a mixed team, we needed an experimental testbed where a robot could perform tasks synchronously with humans. We also required a group activity where each member's actions would have impact on others' actions, as well as have impact on the dynamics of the team overall.

Therefore, we designed a movement task where a team of humans and a robot could coordinate their motion in real-time. Specifically, we explored SJA within the context of synchronous dance. In concert with an experienced dancer, we choreographed a routine to the song *Smooth Criminal* by Michael Jackson, which is in 4/4 time. The dance is iterative, and performed cyclically in a counter-clockwise manner (see Figure 8.1-A). There are four *iterations* in a dance session, corresponding to each of the cardinal directions (North, West, South, and East). Each iteration includes the dancers taking the following steps in order: move forward and backward twice, then, clap, and turn 90-degrees (see Figure 8.2) [166].

### 8.1.1 Data acquisition process

Figure 8.1-A shows the data acquisition setup. Three human participants and a Turtlebot v.2 robot were arranged in two rows. Four Microsoft Kinect v.2 sensors were positioned approximately three feet above the ground at each of the cardinal directions. Each sensor was connected to a computer (client) to capture and process the depth, infrared, and skeletal data from the Kinect. All four clients and the server ran Windows 8 on an Intel Core i5 processor at 1.70Hz with 12GB of RAM.

As we are studying synchronous activity, it was critical all clients and the robot maintained a consistent time reference. Thus, we created a server to manage communication and global time synchronization. Synchronization architecture details can be found in Iqbal et al. ([74]).

Each client performed real-time processing of the raw data in order to detect dance events (e.g., move forward, stop, etc), which it sent to the server with a timestamp. When the server received data from the clients, it generated predictions for how the robot should move using one of two anticipation methods, which are described in Section 9.1. The server was also responsible for determining the *active client*, which refers to which of the four sensors the dancers were facing during a given iteration.

In order to allow for offline analysis, the clients also recorded time-synchronized depth, infrared, audio, and skeletal data using at automated interface with Kinect Studio. The server and robot also kept detailed logs of all communication, odometry information, events received from the active client, and information about the dancers.

### 8.1.2 Client-side data processing

We extracted five high-level events from the participants' movements during the dance: *start moving forward*, *stop moving forward*, *start moving backward*, *stop moving*

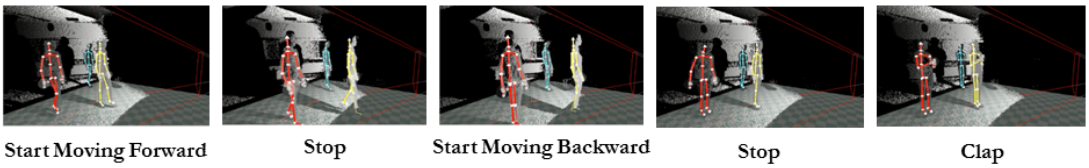


**Figure 8.2**: Five high-level were detected during the dance using skeletal data from participants. One iteration of the dance consists of two passes of the *start moving forward*, *stop moving forward*, *start moving backward*, *stop moving backward* events, and a *clap* event in order.

*backward*, and *clap*. The *start moving forward* event is detected when a participant begins approaching the Kinect, and *stop moving forward* when they stop moving. Similarly, as a participant moves away from the sensor (backward), that is identified as a *start moving backward* event, and when they stop, *stop moving backward*. We also detected participants' *clap* events, which occurred at the end of each iteration. See Figure 8.2.

To detect these events from participants' body movements, clients used the skeletal positions provided by the Kinect. Clients calculated forward and backward motion onsets along the *z-axis* primarily using the spine base body joint position, as it is the most stable and reliable joint position when participants are in motion.

However, there were times when participants did not move their spine base, but did move their mid-spine, shoulders, or neck, to signal the onset of motion. Therefore, clients also used these positions, again along the *z-axis*, to detect four additional events: *early start moving forward*, *early stop moving forward*, *early start moving backward*, and *early stop moving backward*. For these early events, clients calculated joint change positions by comparing the current and previous frame. If at least half of the joint positions changed, then it indicated the participant had started moving. To detect *clap* events, clients used the x and y coordinates from the 3D skeletal position of the left and right hand and shoulder joints. Claps occurred when the ratio of the distance between the hands and the distance between the shoulder joints was less than a threshold (0.6), and when this ratio value reaches a local minima.

### 8.1.3 Robot command generation and execution

After the server determines which movement the robot should make, which it does using an anticipation methods described in 9.1, it sends a movement command to the robot. These commands include: *move forward*, *move backward*, *stop*, and *turn*. The server translated the clap commands into rotation commands while sending it to the robot, since the robot can't clap.

For example, suppose *Dancer* 2 was the most synchronous person during iteration 1, i.e., $MSP(itr_1)$ was *Dancer* 2. *Dancer* 2 performed a *start moving forward* event three seconds from the start of $itr_1$. So, during $itr_2$, it was assumed that the *start moving forward* would happen three seconds from the iteration's start. Thus, if the server received a sufficient number of *early start moving forward* events around $t_3$, then it notified the *robot command generator* to generate commands to execute forward movement. This process was similar for all other regular events, excluding the clap event.

The robot, which ran the Robot Operating System (ROS) version Hydro on Ubuntu version 12.04, accepted commands from the server, parsed the commands, and used an ROS publisher to send movement commands to the controller. The robot is capable of forward and backward movement, and can rotate on its vertical axis in either direction.

## 8.2 Experiments

### 8.2.1 Pilot studies

Before performing the main experiment to compare performance between the two anticipation methods, we performed a series of pilot studies to test the setup of the system, and set various parameters for the two anticipation methods. We conducted two sets of

pilot studies, with a total of seven participants (three women, four men). Participants were opportunistically recruited, and compensated with a $5 gift card for participating [166].

During the first set of pilots, a sole participant danced with the robot. Here, we sought to measure two things: how fast the robot received action messages, and how accurately the robot performed with the human participant.

During the second set of pilot studies, a group of three participants danced with the robot. Here, we sought to establish appropriate parameters for the anticipation methods. To acquire these measurements, we recorded events generated from server logs as well as from odometry data from the robot. We compared the two, noting differences in velocity, distance, and event timings [166].

### 8.2.2 Main experiment

We recruited a total of nine groups (27 participants in total, 3 persons per group) for our main experiment. 14 participants were women, 13 were men. Their average age was 22.93 years (s.d. = 3.98 years), and the majority were undergraduate and graduate students. Only 3 participants had prior dancing experience, 24 did not. Participants were recruited via mailing list and campus advertisement. Upon scheduling a timeslot, participants were randomly assigned to join a group with two others. Each participant was compensated with an $8 gift card for their time.

After giving informed consent, participants viewed an instructional video of the choreographed dance and the experimenters explained the different movements. The participants then had time to practice the dance movements as a group as many times as they wanted. During this practice session, the robot did not dance with them.

Following the practice session, the group participated in three dance sessions. During the first session, only humans participated in the dance. During the last two

sessions, the robot joined the group. In Sessions 2 and 3, the robot moved using either ECA then SIA, or SIA then ECA. (The order was counter-balanced to avoid bias). Participants were blind as to which method was in use.

During the last two sessions, the four clients recorded depth, infrared, and skeletal data of the participants, and the server logged all event and timing data. A single camera mounted on a tripod recorded standard video of the experiment for manual analysis purposes only.

Following the experiment, participants completed a short questionnaire asking them to rate which of the two dance sessions they felt was more synchronous, a measure we have used in prior work [82]. Participants also reported which session they felt they were more synchronous with the rest of the group.

## 8.3 Robot data pre-processing

The server provided the human movement data logs and the clients raw data during the experiment, as detailed in Sections 8.1 and 9.1.1. However, to conduct a complete comparison between the two anticipation methods, it is also necessary to determine how and when the robot actually moved during the two experimental sessions. To do this, we used the timestamped odometric data from the robot ($x$ and $y$ pose and $angular-z$ orientation), as well as the server-side robot command logs.

We calculated the same events for the robot as for the humans (forward, backward, stop, turn). Based on the changes in two consecutive $x$ or $y$ pose values and the robot's heading, we calculated whether the robot was moving forward or backward. For example, when the robot faced the first Kinect sensor and moved forward, then the changes in two consecutive pose values would be positive, if moving backward, negative. We detected turn events using changes greater than 0.4 in the $z$ heading value of the Turtlebot's

*angular twist* message. (Note, *turn* events are considered equivalent to the humans' *clap* events in our analysis.).

*Stop* events were determined when a difference less than 0.002 was detected between two consecutive poses. These stop events were classified as forward or backward depending on the heading of the robot.

After detecting all events for the robot, we manually checked the data files for any errors. During this process, we determined a 7% missing event rate. These missing events were evenly distributed across both of the anticipation methods. We manually checked the recorded video and odometric logs from the robot, and determined the robot actually moved correctly during the majority of those instances, so manually inserted the missing events into the files. There were a few instances (about 3.7% overall) when the robot did not perform the activity that it was instructed to perform, which was mostly due to network latency. We discarded those data from the analysis.

## 8.4 Data analysis and results

To compare the performance and accuracy of the two anticipation methods, we first measured how synchronously the entire group, including the robot, coordinated their movements during both sessions. We then measured how appropriately timed the robot's motion was with its human counterparts.

### 8.4.1 Measuring synchronization of the group

Using the method described in [82] and discussed in Section 7.2, we measured the degree of synchronization of the group for each iteration of the dance. First, we created individual time series for each of the dancers and the robot. Events in the time series were *start moving forward*, *stop moving forward*, *start moving backward*, *stop*

*moving backward*, and *clap*). Then, we calculated the pairwise synchronization index for each pair using the method described in Section 7.2.1.

From the pairwise synchronization index, we built a group topology graph ($GTG$) and calculated the individual synchronization index for each human dancer, as described in Section 7.2. As the humans physically stood very close in proximity, we assumed that each of the group members was influenced by all other members of the group across the course of an entire dance session. (Every iteration, participants rotated their position, so a person in the front at $itr_i$ will end up in the back by $itr_{(i+2)}$.) Thus, in the analysis every human was connected in the graph with all other members of the group, including the robot.

When calculating the robot's individual synchronization index, we employed slightly different analyses between ECA and SIA. For ECA, because the robot's motion was based on the average of all dancers' motions in the previous iteration, when building the $GTG$ all edges from the robot connected to all other human group members. However, for SIA, at any given $itr_i$ the robot was only ever following $MSP(itr_{(i-1)})$ in real time. Thus, during $itr_i$ the robot was only influenced by that person, not by the other group members. Thus, it is logical to take only the pairwise synchronization index between the robot and that person into account while calculating the individual synchronization index of the robot and building the $GTG$ for that iteration. Therefore, we only considered an outgoing edge from the robot to $MSP(itr_{(i-1)})$ in the $GTG$.

After measuring the individual synchronization index, we calculated the group synchronization index for each group using the method described in [82]. Here, we describe the method very briefly.

While calculating the group synchronization index, both the individual synchronization index as well as the members' connectivity to the group was taken into consideration. For a given vertex in the $GTG$, the ratio of the number of outgoing edges

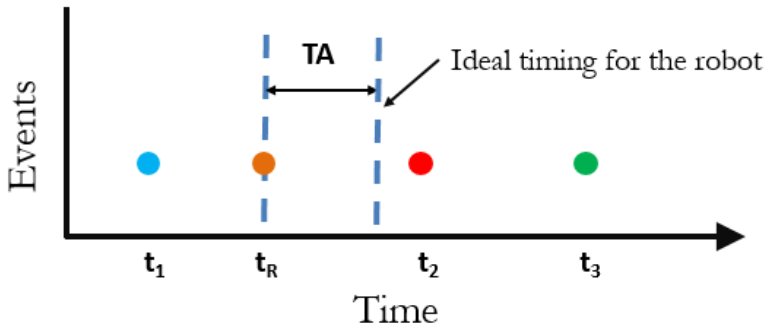


**Figure 8.3**: Timing appropriateness calculation for the robot's movement.

connecting to it, and the number of maximum possible edges in a very synchronized condition for that vertex, is called *the connectivity value* (*CV*). Thus we can define *CV* of series $s_i$ as:

$$CV(s_i) = \frac{\sum_{j=1,\ldots,H,\ j\neq i} f(s_i, s_j)}{H-1} \tag{8.1}$$

The *CV* represents how well an individual is synchronized with the rest of the group. First, we calculate each individual's synchronization index multiplied by their *CV*. Then, the overall group synchronization index is computed by taking the average of this product [82]. So, the overall group synchronization index, $G_\tau$, is computed by:

$$G_\tau = \frac{\sum_{i=1,\ldots,H} I_\tau(s_i) \times CV(s_i)}{H} \tag{8.2}$$

While calculating the group synchronization index, we used $\tau = 0.25s$. This value means we considered two events synchronous when the same types of events in two time series occurred within 0.25 seconds of one another.

Table 8.1 presents the group synchronization indices (GSI) for each group (three humans and one robot), across both anticipation methods (ECA and SIA), and across the four iterations per session). The table also presents the average GSI for each group in the rightmost column. Boldface is used to indicate which of the two methods yielded a

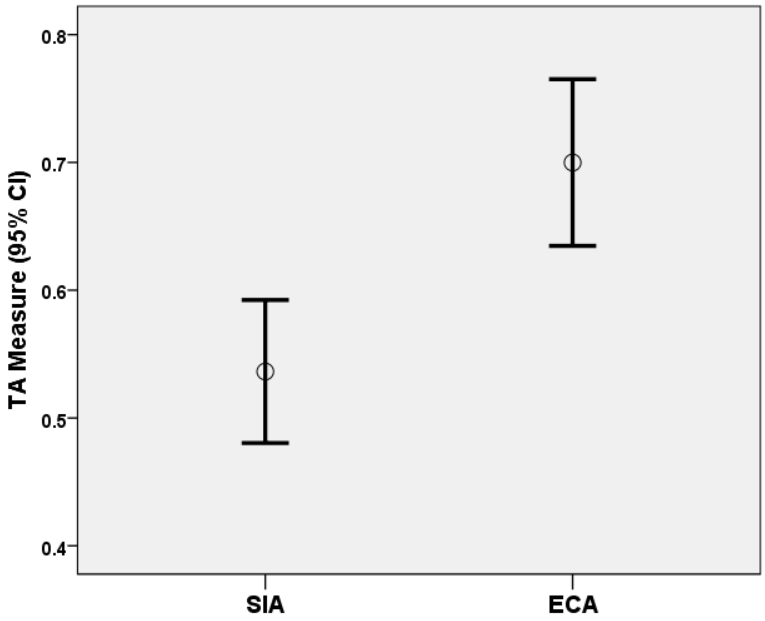


**Figure 8.4**: Timing appropriateness measure for the robot with 95% confidence interval for both methods, SIA and ECA.

higher GSI, and this is indicated for both the per-iteration GSI and the average GSI per group.

For 22 out of 36 total individual dance iterations, the SIA method yielded a higher GSI than the ECA method. And in 7 out of 9 trials, the SIA method yielded a higher GSI than the ECA method.[1]

Using a discrete analogue scale, we asked participants to rate on a scale from 1-5 how synchronous they thought the robot was with the other humans during the sessions. Based on their responses, we measured the more synchronous session of that trial, for which 2 out of 3 dancers agreed on their rating. For 7 / 9 trials, this collective rating matched with the more synchronous session of the trials determined by our method (See Table 8.1, last two columns.)

[1]Note, due to a small sample size ($n$ = 36), it would be dubious to run statistical means comparisons, and one should not accept a *p-value* with certainty [51]. Instead, we agree with Gelman [51] that reliable patterns can be found by averaging, as reported here.

**Table 8.1**: Group Synchronization Indices (GSI) for All Groups. Each group includes three people and one robot.

| Group No. | Iteration No. | GSI | | Mean GSI | |
|---|---|---|---|---|---|
| | | ECA | SIA | ECA | SIA |
| 1 | 1 | 0.39 | **0.40** | 0.29 | **0.45** |
| | 2 | **0.30** | 0.26 | | |
| | 3 | 0.15 | **0.63** | | |
| | 4 | 0.33 | **0.52** | | |
| 2 | 1 | 0.28 | **0.66** | 0.46 | **0.56** |
| | 2 | **0.39** | 0.37 | | |
| | 3 | **0.63** | 0.51 | | |
| | 4 | 0.54 | **0.71** | | |
| 3 | 1 | 0.37 | **0.40** | 0.45 | **0.52** |
| | 2 | 0.48 | **0.52** | | |
| | 3 | 0.59 | **0.62** | | |
| | 4 | 0.37 | **0.55** | | |
| 4 | 1 | **0.30** | 0.20 | **0.33** | 0.29 |
| | 2 | 0.37 | **0.38** | | |
| | 3 | **0.37** | 0.30 | | |
| | 4 | **0.28** | 0.28 | | |
| 5 | 1 | **0.31** | 0.24 | 0.38 | **0.43** |
| | 2 | 0.39 | **0.44** | | |
| | 3 | 0.52 | **0.59** | | |
| | 4 | 0.30 | **0.46** | | |
| 6 | 1 | **0.50** | 0.43 | 0.41 | **0.41** |
| | 2 | 0.41 | **0.45** | | |
| | 3 | **0.40** | 0.32 | | |
| | 4 | 0.33 | **0.46** | | |
| 7 | 1 | **0.41** | 0.25 | 0.42 | **0.47** |
| | 2 | **0.42** | 0.29 | | |
| | 3 | 0.35 | **0.66** | | |
| | 4 | 0.52 | **0.68** | | |
| 8 | 1 | **0.78** | 0.32 | 0.50 | **0.52** |
| | 2 | 0.40 | **0.48** | | |
| | 3 | 0.41 | **0.72** | | |
| | 4 | 0.41 | **0.56** | | |
| 9 | 1 | **0.56** | 0.35 | **0.45** | 0.44 |
| | 2 | **0.35** | 0.34 | | |
| | 3 | 0.42 | **0.46** | | |
| | 4 | 0.46 | **0.59** | | |

### 8.4.2 Measuring robot timing appropriateness

For both anticipation methods, we aimed to have the robot perform its actions (events) as close as possible in time to its human counterparts. Thus, we measured how close the robot's actual movement was to what the humans were doing at that time.

Thus, as a measure of timing appropriateness of the robot, we calculated the absolute time difference between the time when the robot performed an event, and the ideal timing of that event. As a measure of the ideal timing of an event, we took the average timing of an event performed by the humans. This measure is similar to the absolute offset measure used in [69], however, the timing appropriateness measure used here is within the context of a group.

First, we represented all events associated with the humans during an iteration by a time series. Then, we clustered all the similar types of events together with those that were performed by the dancers within a time threshold, ε. For example, for a single event $e$, we calculated the timing of the event performed by three human participants within ε, i.e., $t(dancer_1, e)$, $t(dancer_2, e)$, $t(dancer_2, e)$. We also calculated the timing of that event performed by the robot, $t(robot, e)$. Then, to calculate the ideal timing for the robot, we take the average of these times of this event performed by the humans. Thus, $t(robot_{ideal}, e) = (t(dancer_1, e) + t(dancer_2, e) + t(dancer_3, e))/3$. Then, we calculated the timing appropriateness ($TA$) of that event performed by the robot as, $TA(e) = |(t(robot, e) - t(robot_{ideal}, e))|$. Figure 8.3 presents an example calculation of $TA$ for event $e$.

After calculating $TA$ for each event during all the trials, we created two histograms, one for each anticipation method. We used a bin size of 0.1 seconds, starting at $0s$ and going to $2.5s$. Then, we calculated the frequency of the events for which the $TA$ falls within that time span.

In Figures 8.5-A and B, we present histograms representing the timing appropri-

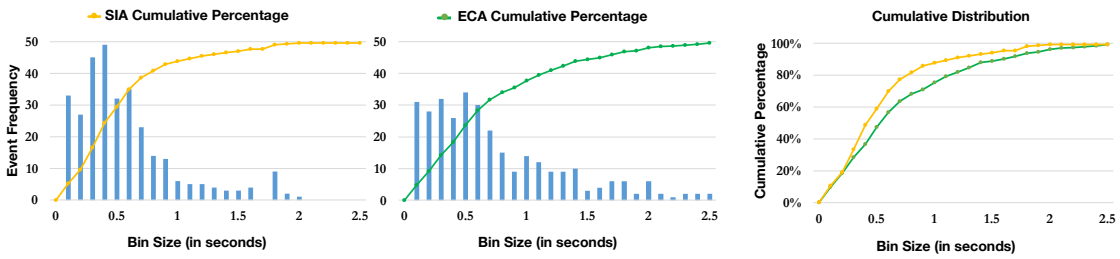


**Figure 8.5**: Event frequency distribution and the cumulative percentage distribution of the timing appropriateness measure for the two anticipation methods. SIA (left) and ECA (right). The rightmost graph shows the distribution of the timing appropriateness measure for both methods.

ateness measure, and the cumulative percentage of event frequencies, for the ECA and SIA methods respectively. Figure 8.5-A (ECA), shows that the robot was able to perform 81.88% of its events within 1.2$s$, and 90% of its events within 1.6$s$ of the appropriate timing. Figure 8.5-B (SIA) shows that the robot performed 81.65% of the events within 0.8$s$, and 90.82% of the events within 1.2$s$ of the appropriate timing.

Figure 8.5-C presents the cumulative percentage of events for both methods together. One can find that the robot performed the events more appropriately during the SIA method, than compared to the ECA method.

For the SIA method, the mean for the timing appropriateness measure was 0.54$s$ (s.d. = 0.59$s$) (See Figure 8.4). For the ECA method, the mean timing appropriateness measure was 0.70$s$ (s.d. = 0.50$s$) (See Figure 8.4). While these data did not have a normal distribution, as is visible from the graph and a normality test, they did have a sufficient number of means to compare statistically. We conducted a Wicoxon Signed Rank Test, and found that the timing appropriateness values for the ECA method were significantly larger than for the SIA method, $z = -4.399, p < 0.05, r = -0.18$. This means that when using the SIA method, the robot moved more appropriately in time than when using the ECA method.

## 8.5 Discussion and future work

The results suggest that the human-robot team was more synchronous using SIA than using the ECA method. Moreover, when SIA was used, the robot was able to perform its actions significantly closer to the appropriate timing of the event. This supports the idea that SIA is well-suited to provide movement coordination information to a robot during an SJA scenario.

Additionally, these results might support the robustness of the SIA method over the ECA method, as the SIA method is more dynamic and adaptable to change within the group. In our study, the SIA method chose the most synchronous dancer in the group, and used that to inform the robot's actions in real-time. However, relying on a method like ECA would mean that if a dancer was moving asynchronously within the group, the robot's motion could be adversely affected (as it is following everyone). SIA is robust to handle this phenomenon, as a person who performed asynchronous movements within the group is unlikely to ever be chosen as the most synchronous person.

This work shows that taking team dynamics into account can be useful for robots when conducting coordinated activities with teammates. This work can lead others in the robotics community towards further investigating the role of a group on behavior, rather than just focusing on individuals. This has implications not only for human-robot interaction, but also for multi-robot systems research. We are currently exploring the effect of different anticipation methods in multi-human multi-robot scenarios [83, 77].

One limitation of this work is how event detection is calculated. In the current setup, a predefined set of human activities were detected by the system to understand the team dynamics. Building on this foundation, our future work will include incorporating human gross motion directly to the synchronization measurement step, instead of using pre-labelled events. Moreover, we are also planning to incorporate a decision module for

robots, which will use the perceived knowledge to select the best decision from a set of options, based on the context [147, 151].

Another limitation of the current method is how it uses team metrics, and task-related information. For example, the method does not yet incorporate dancer expertise, nor does it factor in the tempo or dynamics of the music. In the future, we plan to incorporate an understanding of these factors. For example, in a team of novice dancers, a robot could perhaps keep a team on tempo.

Movement coordination is an important, emerging research area in robotics, neuroscience, biology, and many other fields [50, 8, 229, 107, 152, 195]. Our work helps enable robots to have a better understanding of how to coordinate with the environment. This can be useful both for solving problems in robotics, and perhaps also in fields beyond.

## 8.6 Chapter summary

In this chapter, we validated the two anticipation methods within a human-robot synchronous dance scenario. We compared the results of the SIA method with the ECA method which did not rely on team dynamics. The results suggest that the SIA method performed better both in terms of more closely synchronizing the robot's motion to the team, and also exhibiting more contingent and fluent motion. These findings suggest that the robot performs better when it has an understanding of team dynamics than when it does not. In the next chapter, we will investigate how the presence of robots affects group coordination when both the anticipation algorithms they employ and their number (single robot or multi-robot) vary.

## 8.7 Acknowledgements

I want to thank Samantha Rack for helping me designing and implementing the client-server architecture, setting up the experimental testbed, and during the experiments.

This chapter contains material from "Will you join the dance? Toward synchronous joint action in human robot teams," by T. Iqbal., S. Rack, and L. D. Riek, which appears in Proceedings of the Conference of Joint Action Meeting, 2015 [78], and "Movement coordination in human-robot teams: A dynamical systems approach," by T. Iqbal., S. Rack, and L. D. Riek, which appears in IEEE Transactions on Robotics, 2016 [79]. I was the primary investigator and author of these papers.

# Chapter 9

# Coordination dynamics in multi-human multi-robot teams

In Chapters 7 and 8, we explored human-human and human-robot team dynamics within the context of robots sensing and responding to human teammates. We also compared the two anticipation algorithms (SIA and ECA) on several dimensions, such as how appropriate the timing of events were by the robot. We found that when the robot took team dynamics into account, it performed more appropriately with the rest of the group.

In this Chapter, we want to explore how both the human and overall team dynamics change within multi-robot scenarios. We are motivated to explore several research questions in the context of team dynamics.. The first two questions concern the introduction of robots into a human-human teamwork scenario - what are the effects of adding one or two robots to a group of humans, both on their coordination with each other and the group as whole? The next two questions ask, how do different anticipation algorithms on robots affect human behavior? Might algorithms more sensitive to human team dynamics affect motion differently?

To address these research questions, we have extended the experiment described in Chapter 8, and designed a human-robot interaction scenario, where one or two autonomous mobile robots observes a group of human dancers, and then successfully and contingently coordinates their movements to "join the dance" based on different scenarios. We employed the two methods (described in Chapter 7) in order for the robots to coordinate their movements with the human group. One method is a synchronous joint action based anticipation method (SIA) which takes team dynamics into account, while the other method (ECA) did not rely on team dynamics.

We conducted an experiment with four conditions: 1) only three humans performed a synchronous dance in a group, 2) one robot joined a group of humans to perform a synchronous dance, 3) two robots joined the group of humans, and the same movement anticipation and generation method was used for the both robots, and 4) two robots joined the humans, however, different movement anticipation and generation methods were used by the robots. During all the interaction scenarios, the humans could either directly see or overhear the sound of the robot's movements.

## 9.1 Experimental setup

As the testbed for this experiment, we used the same synchronous dance scenario that we used in Chapter 8. However, based on different the experimental scenarios, during this experiment a group of robots coordinated their motion with a group of humans.

Just to recall, in concert with an experienced dancer, we choreographed an iterative dance routine to the song *Smooth Criminal* by Michael Jackson, which is in 4/4 time. There are four *iterations* in a dance session, and is performed cyclically in a counter-clockwise manner. The group performs this dance facing each of the cardinal directions (North, West, South, and East) during an iteration. Each iteration includes the


**Figure 9.1**: In the study, three participants danced together across three conditions, A) humans alone, B) humans with one robot, and C) humans with two robots. In B) and C), there were two variations, where different anticipation algorithms were executed on the robot.

dancers taking the following steps in order: move forward and backward twice, clap, and a 90-degree turn.

The client-side data processing was the same that we used in Chapter 8. However, if there were two robots in the group, then the server generated movement commands for both of the robots based on the experimental scenarios.

### 9.1.1 Main experiment

To explore the effects of visual and auditory feedback from the robots during an intentional coordination task, we physically incorporated robot(s) in the group in such a way that participants were able to hear the robot's motors at all times, and view a robot during some iterations (See Fig 9.1). This scenario provided the opportunity to investigate the effect of robot motion (including auditory, and visual feedback) on the group's coordination.

We also provided an external rhythmic signal to participants to help them to maintain a consistent, synchronized tempo during during the dance. Participants were instructed to maintain awareness of the other participants' and robots' movements, and

to dance synchronously as a group.

We recruited a total of seven groups for our main study, with three people per group. Data of one group were excluded due to the robot losing connectivity with the server, so here we report the results from six groups (18 participants in total). 11 participants were women, 7 were men. Their average age was 24.7 years (s.d. = 4.5 years), and the majority were undergraduate and graduate students. Participants were recruited by word-of-mouth. Upon scheduling a time slot, participants were randomly assigned to join a group with two others. Each participant was compensated with an $10 gift card.

After giving informed consent, participants viewed an instructional video of the choreographed dance and the experimenters explained the different movements. The participants then had time to practice the dance movements as a group as many times as they wanted. During the practice session, the robot did not dance with them. Following the practice session, the group participated in six dance sessions, in four phases.

1. *Phase 0*: During the first phase of the dance sessions, only the humans participated in the dance.
2. *Phase 1*: During the second phase of the dance sessions, one robot joined the group. The dancers participated in two dance sessions during this phase, where the robot moved using either ECA then SIA, or SIA then ECA. The order was counterbalanced to avoid bias within this phase.
3. *Phase 2*: During this phase, two robots joined the dance with the humans. The dancers also participated in two dance sessions during this phase, with either SIA or ECA methods controlling the robot's movements. The order was counterbalanced to avoid bias within this phase.
4. *Phase 3*: During the last phase, two robots moved with the humans. However, movements of one robot were generated using ECA, and movements of the other

robot were generated using SIA. The anticipation methods for the robots were counterbalanced to avoid bias.

We controlled for several factors relating to how participants altered their behaviors relative to the robots' motion in several ways. First, we provided an external rhythmic signal to the human participants to help them maintain a consistent tempo. Second, we deliberately choreographed an easy dance (walking, turning, and clapping), and gave substantial practice time before the experiment began. This enabled participants to develop their muscle memory, so that they would not be easily distracted during the experiment. Third, because we had six counterbalanced conditions (i.e., random ordering per phase for each participant group), any effects relating to the robots being distracting will be greatly lessened.

During all sessions, the clients recorded depth, infrared, and skeletal data of the participants, and the server logged all event and timing data. A single camera mounted on a tripod recorded standard RGB video of the experiment for manual analysis purposes only. Following each session, participants completed a short questionnaire asking them to rate in a discrete visual scale how well-synchronized the group was during that dance session.

## 9.2 Analysis and results

To address the first two research questions, we need to measure how well coordinated the human participants' movements are when we consider them separate from the group, without the robots. In order to address the research questions three and four, we need to measure how well coordinated the whole group are including both the humans and the robots. To measure the synchronization of the group, we used the same method described in Chapter 8.

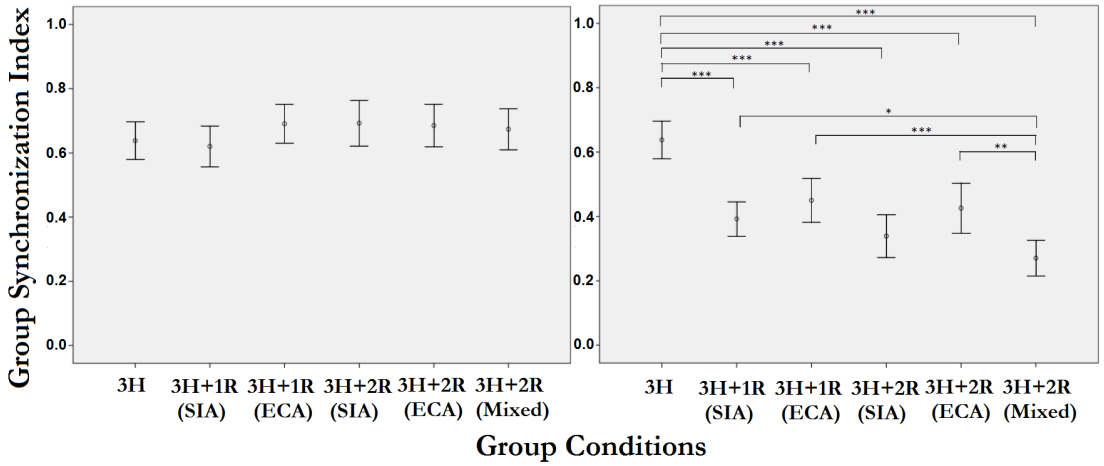


**Figure 9.2**: The group synchronization index (GSI) values for different group conditions along x-axis, where 3H means 3 humans, 1R means 1 robot, 2R means 2 robots, SIA means synchronization index based anticipation, and ECA means event cluster based anticipation. The graphs on the left shows the GSI values of the human group, and the right shows the GSI values of the whole group. The significant difference in GSI values between groups are shown in * (* = $p < 0.05$, ** = $p < 0.01$, and *** = $p < 0.001$)

To address our research questions, we measured both the group synchronization index only considering the human participants (GSI(H)) across all experimental sessions, as well as the group synchronization index of the whole group (GSI(G)). First, we analyze the effect on the group synchronization index only considering the human participants for different experimental scenarios. Then, we analyze the effect on the whole group synchronization index for different experimental scenarios.

### 9.2.1 Effect on the GSI(H) values

We conducted an one-way repeated-measures ANOVA with the Bonferroni correction on the human group synchronization index (GSI(H)) values of all six experimental sessions, consisting of one session of *Phase 0*, two sessions of *Phase 1*, two sessions of *Phase 2*, and one session of *Phase 3*.

Mauchly's test indicated that the assumption of sphericity had been met, $\chi^2(14) =$

$9.38, p > 0.05$. One-way repeated-measures ANOVA with the Bonferroni correction indicated that the human group synchronization indices were not significantly different across all experimental conditions, $F(5,115) = 1.37, p > 0.05, \omega^2 = 0.03$. These results suggest that the human group synchronization index was not affected by adding one or more robot as a performer to the group, irrespective of the anticipation algorithms. Fig 9.2 shows errors bars of GSI(H) values across all experimental conditions.

### 9.2.2 Effect on the GSI(G) values

We conducted an one-way repeated-measures ANOVA with the Bonferroni correction on the whole group synchronization index (GSI(G)) values of all six experimental sessions, consisting of one session of *Phase 0*, two sessions of *Phase 1*, two sessions of *Phase 2*, and one session of *Phase 3*.

Mauchly's test indicated that the assumption of sphericity had been met, $\chi^2(14) = 18.56, p > 0.05$. One-way repeated-measures ANOVA with the Bonferroni correction indicated that the group synchronization indices (GSI(G)) across the experimental conditions were significantly different, $F(5,115) = 22.59, p < 0.05, \omega^2 = 0.21$.

These results also suggest that the GSI(G) values of the *Session*1 were significantly different than all other experimental conditions (for all conditions $p < 0.001$). This result indicates that there is a change in the degree of the group synchronization when we added one or more robots to the group, regardless of the anticipation algorithms.

The results also indicate that the GSI(G) values of the *sessions* of *Phase*1 were not significantly different than the *sessions* of *Phase*2 (for all conditions $p > 0.05$). It suggests that there is no significant effect on the GSI values when we add an additional robot to a three human and one robot group of the same robot behavior, regardless of the robot anticipation algorithms.

However, the results also indicate that the GSI(G) values of the *sessions* of

*Phase*1 were significantly different than the *session* of *Phase*3 ($p < 0.05$ for SIA method of *Phase*1 and *Phase*3, and $p < 0.001$ for ECA method of *Phase*1 and *Phase*3). This suggests that there is a significant effect on the GSI values when we add an additional robot with different behavior to the group, regardless of the robot anticipation algorithms.

Our results also indicate that the GSI(G) values of the *session* of *Phase*2 when the SIA algorithm was used were not significantly different than the *session* of *Phase*3 ($p > 0.05$). However, the GSI(G) values of the *session* of *Phase*2 when the ECA algorithm was used were significantly different than the GSI values of the *session* of *Phase*3 ($p < 0.01$). This suggests that there is no significant effect on the GSI values when the both robots were performing SIA and when the robots performed a mixed behavior. On the other hand, there is a significant effect on the GSI values when the both of the robots were performing ECA and when the robots performed a mixed behavior. Fig 9.2 shows the errors bars of GSI(G) values across all experimental conditions.

## 9.3 Discussion

To our knowledge, intentional coordination tasks have not been explored in the context of multi-human, multi-robot group interaction scenarios. Our study explored how robots might change this dynamic in intentional group coordination. Our results indicate that heteronegenous behavior of robots in a multi-human multi-robot group have a significant impact on the overall group coordination. This is an important finding, because this indicates that the way the robots move in a multi-human multi-robot group may directly impact the dynamics of the whole group, which raises an important concern about how we must design robots to perform along with humans in coordination to achieve common goals.

Our statistical analysis indicates that the addition of a second robot with heteroge-

neous behavior (*Phase 3*) significantly reduces the group coordination over a single robot condition (*Phase 1*). Similarly, the analysis suggests that an addition of a robot to the human-only group also significantly reduces the group coordination over the human-only group (*Phase 1* vs. *Phase 0*). This is an important finding, because the addition of a robot with same behavior does not change the group coordination significantly (*Phase 1* vs *Phase 2*, for both algorithms). These results might suggest that an addition of a robot with heterogeneous behavior to a group significantly reduces the overall group coordination, and might be an important indicator of human-robot group dynamics.

Overall, these results indicate that the group coordination is significantly affected when a robot joins a human-only group, and is further affected when a second robot joins the group and employs a different anticipation algorithm from the other robot. However, these effects were not found to be significant when the two robots employed the same anticipation algorithm.

Although participants were overall more synchronous by themselves than with the entire human-robot team, this does not imply that the team was grossly asynchronous, and the robot did not have influence on the human movements. For example, suppose that a robot moved sooner than the participants due to physical factors, such as sensor noise or actuator issues. When this happens, it may still influence the human team members to start moving earlier as well. However, the humans do not have this issue and are able to maintain consistent, real-time temporal adaptation among themselves (c.f. [215]), although the whole human group was deviated from the original rhythm. This is something we plan to explore in depth in our future work.

We can also extend our method to work beyond synchronous activities, such as timed but varied collaborative tasks. For example, a human-robot team working in an industrial setting has specific sequences of activities to perform over time, some of which might be independent, and might not happen synchronously. However, the events must

happen contingently; so we can extend our methods to these scenarios.

This research may be helpful for others in the robotics community in exploring novel concepts that affect group dynamics beyond dyad groups. As a whole, humans have complex social structures, and it is necessary for robots to understand these underlying concepts if they are to become widely accepted. This work also has implications not only for human-robot interaction, but also for multi-robot systems research, such as robot swarms.

Building on this foundation, we want to explore the effect of including multiple types of robots with different expertise levels in a human-robot group to perform both intentional and unintentional coordinated movements. We are also interested to explore how different robot morphologies (like humanoids) might affect group synchrony.

## 9.4 Chapter summary

In this Chapter, I described a study exploring the effect of incorporating one, or more robots with a human group while performing an intentional synchronous task together. With the technological advancements in future, humans will encounter more robots working along with them in the same group. This work will help to make informed decisions while designing tasks where there will be involvements of human and robot together. Building on this foundation, in next Chapter, I will present my proposed work on building models for robots to anticipate and adapt to a group, and validating the models within a human-robot teaming context.

## 9.5 Acknowledgements

I want to thank Samantha Rack for helping me designing and implementing the client-server architecture and setting up the experimental testbed. I also want to thank Afzal Hossain, Olivia Choudhury, Michael Gonzales, and Cory Hayes for helping me during the testing phases.

This chapter contains material from "Human coordination dynamics with heterogeneous robots in a team," by T. Iqbal. and L. D. Riek, which appears in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Pioneers workshop, 2016 [84], and "Coordination dynamics in multi-human multi-robot teams," by T. Iqbal. and L. D. Riek, which appears in IEEE Robotics and Automation Letters, 2017 [87]. I was the primary investigator and author of these papers.

# Chapter 10

# Adaptation and Anticipation Models for Robots

My research up to this point, along with other research in HRI, helps robots to recognize the activities around them, understand team dynamics, anticipate future actions of other team members, and synthesize their motions accordingly. However, there exist many group interaction scenarios where the activities performed by the team members are not only synchronous but also change their tempo over time. People are very efficient in these kind of situations, but robots are not. We want to build robots that can leverage these human skills to interact with people in dynamically changing environments.

Many researchers in the fields of psychology and neuroscience investigated how people effectively time their coordinated actions with others, especially in constantly changing environments [170, 117]. One important aspect of successful coordination is precise motor timing. Sensorimotor synchronization is one of the skills that people employ to achieve this [215, 216].

Sensorimotor synchronization is considered to be a fundamental human skill, which is described as the temporal coordination of an action with events in a predictable

external rhythm. People are skilled at sensorimotor synchronization even with sequences that contain tempo changes [215].

Researchers have investigated the sensory motor synchronization process of people [215, 98, 169, 224, 68, 71, 15]. Many of these investigations included joint finger tapping with external rhythmic signals or with virtual characters. These studies also included joint drumming or joint dancing with other people, or robots.

Researchers have found two main concepts that people employ during coordination with other external rhythms: *temporal adaptation* and *temporal anticipation*. While synchronizing with a regular-paced or tempo-changing external rhythm, biological noise (i.e., motor variance associated with central vs. peripheral processing) can cause variability in the movement timing of humans, and thus can lead to temporal errors [215, 216]. People use temporal *adaptation* mechanisms to compensate for temporal errors.

Moreover, when people synchronize their movements with an external rhythm, an important step is for them to *anticipate* the onset of external events, so that they can start their task early enough to coordinate with that event [215]. During this process, the human brain can extract structural regularities in external rhythmic patterns and predict future event timing based on that pattern [199]. This process helps people to make predictions from the regularity or variation in the tempo of the external signal.

Combining these two concepts together, van der Steen and Keller [215] introduced ADAM, an Adaptation and Anticipation Model. Inspired from computational neuroscience, ADAM combines both adaptation (temporal error correction) and anticipation (predictive temporal extrapolation) together. Using computer simulations and model fitting to human behavioral data, the authors found that both adaptation and anticipation play a major role during sensorimotor synchronization.

Understanding tempo-changing behavior will enable us to build adaptable robots across a range of fields, from manufacturing to aid skilled human workers in assembly

processes, to assistive technologies to help people with disabilities. If robots can make sense of these tempo changes as people do, then it will be possible for them to adapt and adjust their motions while working with people. This will help robots to become functional teammates to people.

Despite the fact that people can perform temporal adaptation and anticipation during the synchronization of their movements, robots still do not know how to take advantage of this concept. In this work, we are interested in exploring how robots can leverage these mechanisms to coordinate with others in a team. We seek to enable more fluent human-robot teaming by developing robust algorithms for robots that are sensitive to tempo changes during group activities.

To achieve this, we developed models for robots inspired by the temporal adaptation and anticipation models proposed by [215] to facilitate human-robot team coordination. We extended these models to work in dyads as well as to work with multiple people in teams. We also experimentally validated each of the algorithms through a rhythmic task. These validation studies will enable robots to fluently coordinate with others.

## 10.1 Adaptation and anticipation algorithm for robots

Drawing inspiration from the processes that people employ during a coordination process, the goal of this work is to build adaptive algorithms to enable robots to coordinate with people, both in dyads and in groups. This goal also includes a validation of these algorithms implemented on a live robot.

Thus, building on the ADAM framework [215, 216], we developed an adaptable module for robots to work in human-robot team. It has four parts: an adaptation module, an anticipation module, a combined module for dyads, and a combined module for groups.

We explain how a robot can use these modules in the following subsections. We then describe how a robot can combine these modules to work with multiple people. Finally, we describe a validation of these modules through a human-robot teaming study.

### 10.1.1 Adaptation module

People use temporal *adaptation* mechanisms to compensate their temporal errors that occur while synchronizing with an external rhythm [171]. These errors are mostly caused by the variability in movement timings. By employing these mechanisms, people try to change their movement timings to be coordinated with an external rhythm.

The adaptive module is based on an information processing approach [215, 216]. This approach uses a linear timekeeper to model this error correction process, and compensates the timing error correction on a cycle-by-cycle basis. In the timekeeper models, the error correction process can be described as a linear autoregressive process.

Two separate adaptive processes have been proposed in information-processing theory [131, 202]. These adaptive processes are phase correction and period correction. In phase correction, only the timing of the next action is adjusted to compensate the asynchrony with a perceived rhythmic signal, but the timekeeper interval remains the same. In the period correction process, the next action timing is adjusted, and the internal timekeeper setting is changed to compensate for the perceived asynchrony. Both processes independently modify the timing of the next action based on a percentage of asynchrony [171]. These processes are described in Figure 10.1.

The timing of the next event $(t_n + 1)$ is based on the current action time $(t_n)$ and a timekeeper $(T_n)$. The current action time $(t_n)$ is compensated with the phase $(\alpha)$ and the period $(\beta)$ correction by the most recent asynchrony $(asyn_n)$. The asynchrony $(asyn_n)$ is measured by taking the difference between the timings of the recent action and the perceived rhythmic signal [215]. Thus, the most recent asynchrony $(asyn_n)$ is

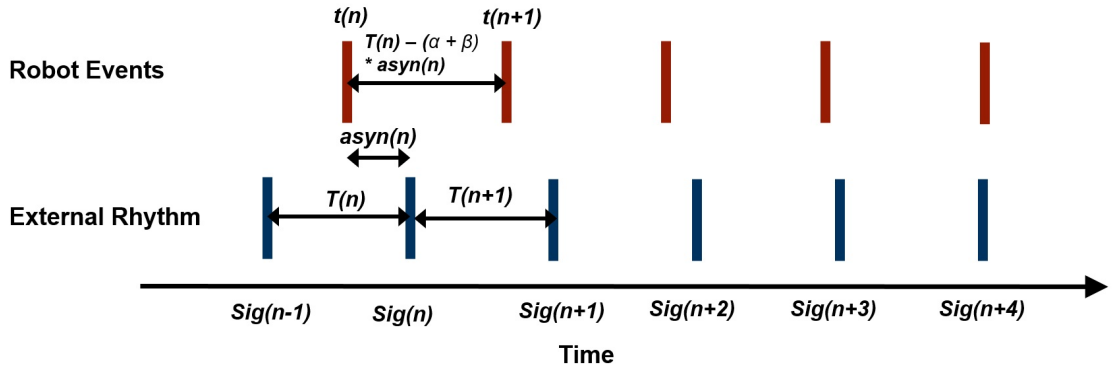


**Figure 10.1**: A schematic representation of conventional sensorimotor synchronization variables and the effect of phase (α) and period (β) correction. Here, a robot event is the tap event and the external rhythmic signal is the tone event. (Adapted from [216])

first multiplied by the sum of the phase correction parameter $(\alpha)$ and period correction parameter $(\beta)$ [215]. Then, the result is subtracted from the current timekeeper period $(T_n)$, and added to the timing of the most recent event $(t_n)$ to determine the timing of the next event $(t_n + 1)$ (See Equation 10.1) [215].

$$t_{(n+1)} = t_{(n)} + T_{(n)} - (\alpha + \beta) * asyn_{(n)} \tag{10.1}$$

In addition, the timekeeper period is compensated for the recent asynchrony and the period correction parameter (See Equation 10.2)

$$T_{(n+1)} = T_{(n)} - \beta * asyn_{(n)} \tag{10.2}$$

Equation 10.1 and Equation 10.2 represent the adaptation process to a perceived rhythmic signal [215].

To implement the adaptation module on robots, the robots need to keep an internal timekeeper. For this purpose, robots observe the activities performed by a person, or the interaction between people, to understand the dynamics in that human-robot team. After detecting the tasks carried out by the participants, the robots measure the average

of inter-onset intervals from the tasks of the participant(s). This value is then set as the initial values of the timekeeper variable. As the interaction progresses, the robots update their timekeeper variables based on the rhythm of the task by using Equation 10.2.

This adaptation module is the reflection of the robot's "self" model. This module only corrects the timing of the next action based on its perceived asynchrony from the rhythmic signal. It does not include a model of perceived rhythmic signals or an internal model of "others."

### 10.1.2 Anticipation module

The anticipation module of the ADAM framework generates a prediction about the timing of the next action so that it coincides with the timing of the next external rhythmic signal [215]. Unlike the adaptation module, the anticipation module makes a prediction of the internal model of "others."

This algorithm utilizes a temporal extrapolation process to generate a prediction about the next timing of an event in the external rhythmic signal. This extrapolation is performed on the inter-event interval of the external signal [215].

Thus, the timing of the next event in the external signal $(Sig_{(n+1)})$ is predicted by adding the extrapolated inter-event interval $(Int_{(n+1)})$ of the rhythmic signal with the timing of the current event $(Sig_{(n)})$. The linear-extrapolation is performed based on the $k$ previous inter-event intervals. Equation 10.3 and Equation 10.4 describe this process [215].

$$Sig_{(n+1)} = Sig_n + Int_{(n+1)} \tag{10.3}$$

$$Int_{(n+1)} = a + b * (n+1) \tag{10.4}$$

Here, the parameters $a$ and $b$ are the y-intercepts and the slope of the line best fitted on the inter-event intervals, respectively .

The predicted behavior $(Pred_{n+1})$ can be represented as the difference between the anticipated timing and the current timing of external events. Thus,

$$Pred_{(n+1)} = Sig_{(n+1)} - Sig_{(n)} \tag{10.5}$$

On the other hand, the tracking behavior $(Track_{n+1})$ represents the interval between the current and the previous external events. Thus,

$$Track_{(n+1)} = Sig_{(n)} - Sig_{(n-1)} \tag{10.6}$$

Now, combining the tracking and the predicted behavior, a robot can anticipate the future timing of an event from Equation 10.7. Here, $m$ represents the prediction-tracking ratio. If $m = 0$, then the anticipation is based only on tracking behavior. On the other hand, if $m = 1$, then the anticipated event timing is totally based on predicted timing.

$$Sig_{(n+1)}(ant) = Sig_{(n)} + (m * Pred_{(n+1)} + (1-m) * Track_{(n+1)}) \tag{10.7}$$

To anticipate timings of future events, a robot detects the actions or events performed by the other team members during an activity following the steps described above. The robot then needs to combine the "others" (anticipation) and the "self" (adaptation) together to predict the future timings of an event. We describe this process in Section 10.1.3.

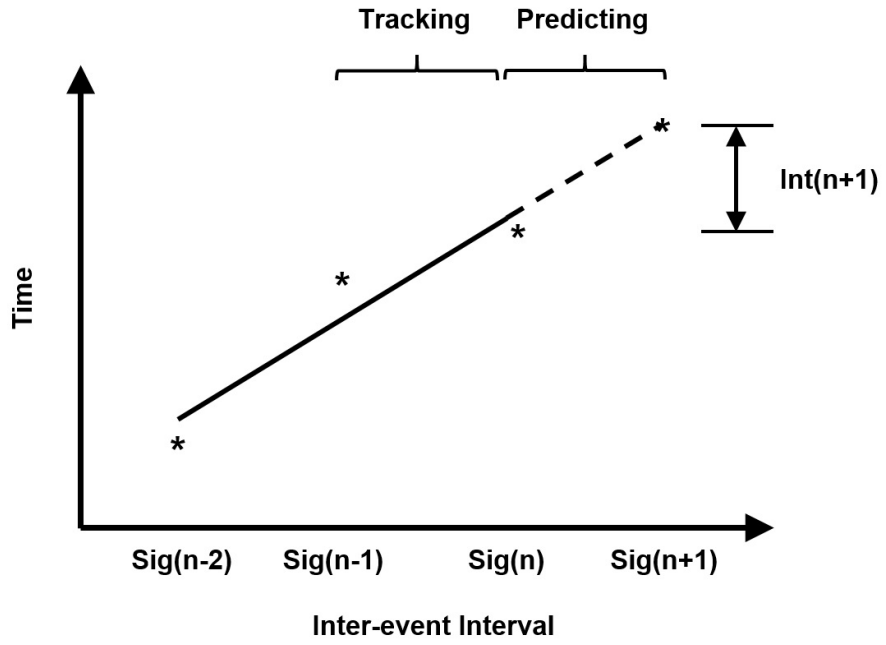


**Figure 10.2**: A schematic representation of the anticipation process. (Adapted from [216])

### 10.1.3 Combined Module: Dyadic

Now, the robot needs to combine the adaptation module and the anticipation module into a predicted timing of the next event. To predict the future timing of an event, the robot needs to measure the asynchrony between the time received from the adaptation module and the time received from the anticipation module. Finally, the next event time is measured by adding this asynchrony with the error corrected time (adaptation) of the event. Here, $\gamma$ represents the anticipatory correction gain.

$$async'_{(n+1)} = t_{(n+1)}(adapt) - Sig_{(n+1)}(ant) \tag{10.8}$$

$$t_{(n+1)} = t_{(n+1)}(adapt) - ((1-\gamma) * async'_{(n+1)}) \tag{10.9}$$

Thus, $t_{(n+1)}$ is the predicted timing of next event, which combines the robot's

"self" and "others" models together. However, this module predicts the event timings of a single human. Now, if the robot needs to interact with multiple people (multiple external rhythms), then the robot needs to combine multiple rhythms together. In the following Section, we described an approach for enabling a robot to accomplish this.

### 10.1.4 Combined Module: Group

The adaptation module corrects the timing of the next action based on its perceived asynchrony from a rhythmic signal. Similarly, the anticipation process can predict only a single external signal. However, in a group setting, the robot might encounter multiple rhythmic signals together.

For example, if a robot is performing a rhythmic coordination task along with people, such as synchronous dancing or drumming, then it perceives different rhythm signals based on how each person performs his/her actions. Thus, a robot needs to combine its knowledge about the perceived signals to inform its decision. We then need to combine the adaptation and the anticipation timings together for the robot to make a decision.

Now suppose that a robot perceives $H$ separate rhythmic signals. Each of these signals can be the timing of the actions performed by each human. For each action, the robot will first need to measure the asynchrony from each perceived rhythm. Then, the robot can combine all these asynchronies by taking an average of these values, as a combined asynchrony. Thus,

$$async_{(n)}(combined) = \frac{\sum_{i=1}^{H} async_{(n)}[i]}{H} \tag{10.10}$$

Therefore, for the group condition we can update Equations 10.1 and 10.2 as:

$$t_{(n+1)} = t_{(n)} + T_{(n)} - (\alpha + \beta) * asyn_{(n)}(combined) \tag{10.11}$$

$$T_{(n+1)} = T_{(n)} - \beta * asyn_{(n)}(combined) \tag{10.12}$$

Similarly, a robot can anticipate the next event timing of a signal using the process described in Section 10.1.2. If we have $H$ participants, the robot also perceives a total of $H$ external rhythmic signals.

Using a similar approach to what we have taken in the adaptation process, we can combine all the anticipation timings by taking an average of the external signal timings.

$$Sig_{(n+1)}(combined) = \frac{\sum_{i=1}^{H} Sig_{(n+1)}[i]}{H} \tag{10.13}$$

Then we can combine the timmings from the adaptation and the anticipation together by using Equations 10.8 and 10.9.

However, a robot can take different approaches to combining multiple signals together during the adaptation and anticipation module. A more sophisticated approach would be to take a weighted average of the asynchronies (for adaptation) or of the anticipated timings of each signal (for anticipation) based on how synchronous they were in previous events.

First, our method will measure the asynchrony among the signals for the last $p$ events. Based on the asynchronies, it will rank the signals for each event. The most synchronous signal will be given the highest rank ($H$), and the least synchronous signal will be given the lowest rank (1). Then, we will accumulate the rank values for each signal and denote that as the rank value for that signal.

Therefore, the unified asynchronies can be calculated by taking the average of the perceived asynchronies from the external signals, weighted by their rank values. So,

$$async_{(n)}(combined) = \frac{\sum_{i=1}^{H}(rank[i] * async_{(n)}[i])}{\sum_{i=1}^{H} rank[i]} \tag{10.14}$$

Similarly, the unified anticipation timing can be calculated by taking the average of the anticipated timing of signals, weighted by their rank values. Thus,

$$Sig_{(n+1)}(ant) = \frac{\sum_{i=1}^{H}(rank[i] * Sig_{(n+1)}[i])}{\sum_{i=1}^{H} rank[i]} \tag{10.15}$$

Another ranking approach would be to rank each rhythmic signal by its *stability* score. Stability of a signal can be measured by the standard deviation of inter-event intervals. The lower the standard deviation value, the more stable the signal will be. Thus, we can also use this value to rank the rhythmic signals, and this can be used in Equations 10.14 and 10.15.

## 10.2 Validation of the Modules

In this work, we are interested in exploring whether the methods that people employ to being synchronous with others are also applicable to robots. To explore this, we implemented the algorithms explained in Section 10.1 to enable a robot to perform synchronous actions with people. As we are particularly interested in testing the validity of these algorithms on a robot, we focused on the implementation of the adaptation module, anticipation module, and combined dyadic module. However, we left the validation of the combined group module (described in Section 10.1.4) for our future work.

We are interested in investigating whether a robot can be coordinated with a person in a tempo-changing environment, at least as good as a person can coordinate his/her actions with another person in that condition.

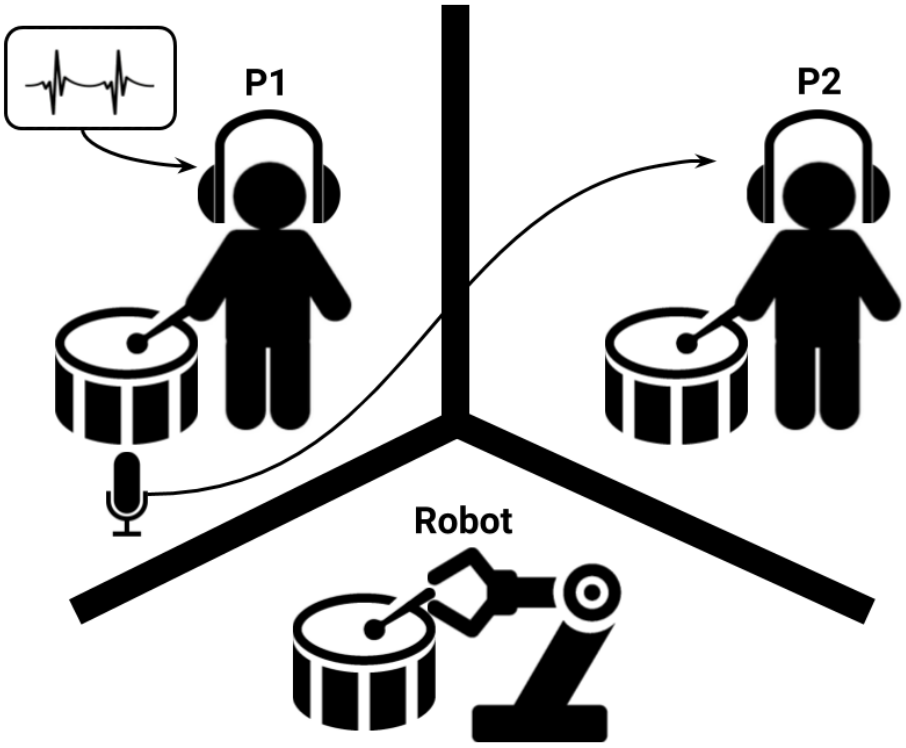


**Figure 10.3**: Experimental setup. The participants and the robot were situated in separate places. There were barriers between them that blocked the vision of each other's drumming. Both participants had sound-blocking headphones on, and could not see the other participants. Person 1 (P1) was listening to an external tempo, Person 2 (P2) was listening to an audio of P1's drumming (via a microphone), and the robot performed following P1.

### 10.2.1 Experimental Testbed

As an experimental testbed, we developed a human-robot drumming scenario. Figure 10.3 shows the experimental setup. In this setup, two people drummed together synchronously. Additionally, a robot observed the participants and joined by synchronously drumming with them.

In this experiment, we were particularly interested to investigate whether a robot can drum synchronously at least as good as a person, when that person only relies on an audible signal to perceive the rhythm.

In this testbed, both participants wore sound-blocking headphones. These reduced the outside noise (such as the drumming sound of other participants), and helped participants concentrate on their primary rhythm.

During the experiment, *Person 1* listened to an external rhythm through his/her headphones. This external rhythm was a metronomic signal. *Person 1* was instructed to follow that rhythm and drum on the beats. This metronomic signal changed tempo based on experimental conditions.

*Person 2* needed to drum synchronously with *Person 1*, only by listening to the drumming of *Person 1*. We placed a microphone underneath *Person 1*'s drum, and this microphone was directly connected to *Person 2*'s headphones. Although *Person 2* could hear other noises, he/she was instructed to ignore other sounds and to drum only by listening to the drumming of *Person 1* on their headphones.

At the same time, a robot drummed synchronously with *Person 1*. Neither of the participants could see the robot drumming. The robot observed the drumming events of *Person 1* from sensors attached to the drum, and drummed synchronously by employing the algorithms described in Section 10.1. We also reduced the sound of robot's drumming by using soft cloths on the robot's drum.

Figure 10.3 shows an experimental condition where the participants are drumming with a robot. Here, two people stood near two tables, and a barrier visually blocked them. There was a drum pad and a drumstick in front of each participant. From each of their respective locations, they could not see the other participant or the robot drumming. Their sound-blocking headphones also reduced any distraction from their assigned task.

### 10.2.2 Robot

We built a robot capable of drumming with a human group (see Figure 10.4). This robot can move a drumstick up and down at a rate of 4-5 Hz to hit a drum pad.

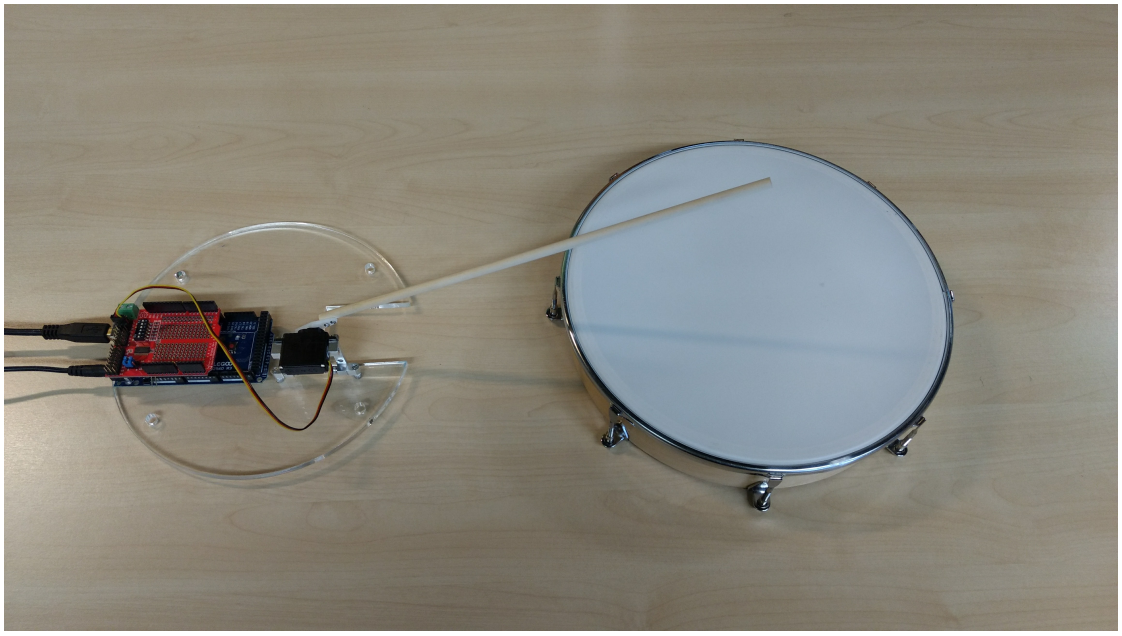

**Figure 10.4**: The first prototype of the drumming robot. It has an Arduino Mega microcontroller, a Renbotic Servo Shield, and servo motor attached to the drumstick.

This rate is very close to how fast an average person (non-drummer) generally can drum, which is at 5-7 Hz [48]). Thus, this robot was well-suited to our experiment.

The robotic arm (the drumstick) was attached to a servo motor. The servo motor was attached to a base for stability. Also, the servo motor was connected to an Arduino Mega and a Renbotic Servo Shield. This Arduino communicated with a computer via rosserial, and was used as an ROS node [136]. We called this node *ArduinoNode* (See Figure 10.5).

The adaptation, anticipation, and combined dyadic module were implemented on ROS. We called this node *AlgoNode*. These ROS nodes communicated to the Arduino via ROS message passing protocol (i.e., rostopic).

Whenever a person or a robot hit a drum pad, the piezoelectric sensor generated a *hit* event. Based on the event timing associated with the *hit* events, the *AlgoNode* generated appropriate timings for the robot to drum. This generated time was sent to the *ArduinoNode*. After receiving this message, the *ArduinoNode* generated the drumming

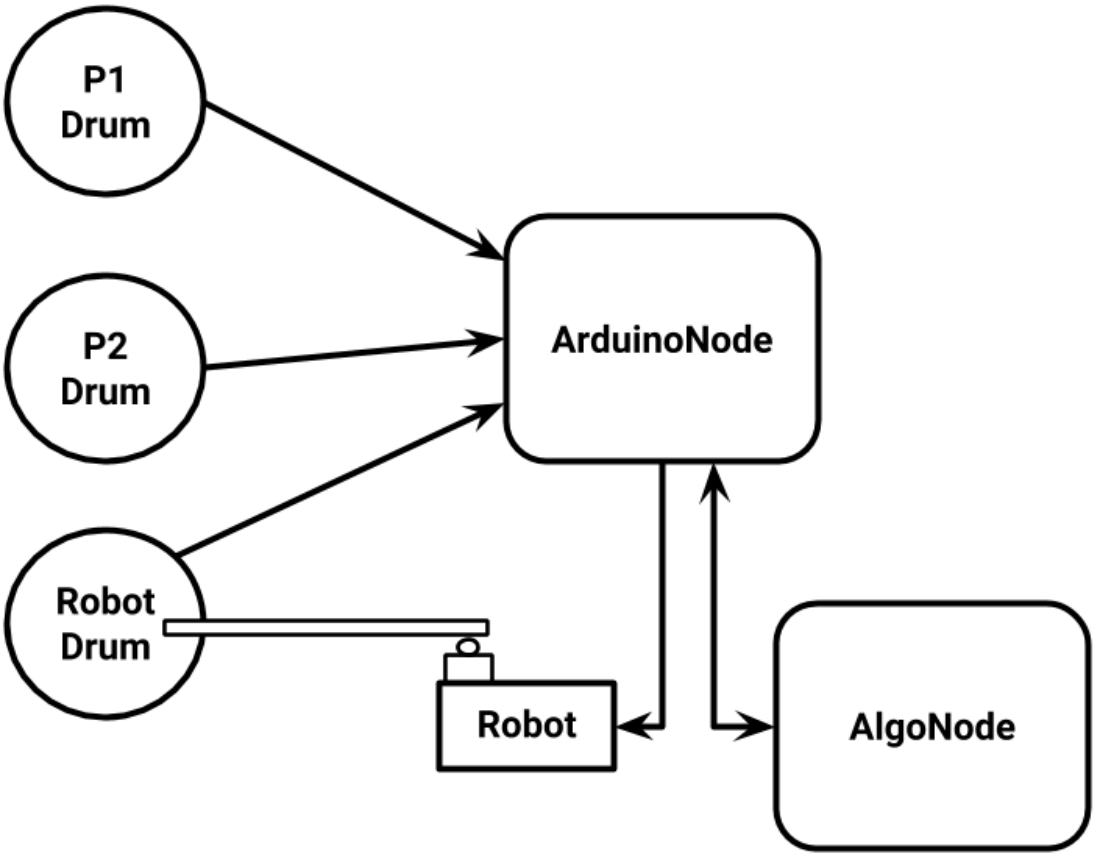


**Figure 10.5**: Different ROS nodes of the system. The arrows denote the connectivity and the ROS message passing direction.

at the exact time.

Figure 10.5 presents the connectivity and the ROS message passing directionality among various components of the system. We discuss this process in detail in Section 10.2.2.

### Drumming Activity Detection

To track the human and robot drumming events precisely, we attached a piezoelectric sensor under each drum pad. These sensors were attached to an Arduino microcontroller. Whenever the participants or the robot hit the drum pad, the Arduino generated a *hit* event. We implemented an Arduino ROS node, which published a *hit* event with this

event time, as well as the drum corresponding to the participant instantaneously. These event times were used by the robot to decide its timing for drumming.

**Robot Action Generation**

After receiving the events from the drums, the *AlgoNode* generated the next action time for the robot. Based on different experimental conditions, we used different algorithms. The *AlgoNode* used that algorithm to generate the timing of the next *hit* for the robot.

At the beginning of each experimental session, *Person 1* drummed for eight beats. During this time, the robot generated the timekeeper $(T_n)$ intervals from the human events; then the robot started to perform with the group.

For next five beats, the robot performed along with Person 1 using only the anticipation' model. The robot started performing using different algorithms starting from the 18th beat of the metronome signal.

After receiving the timing to hit the drum, the *ArduinoNode* started early enough to perform the drumming action on time. We define this time as *earlyStartTime et*. From pilot experiments, we found that for our setup, the robot needs approximately 70 ms to perform a drumming action. Thus, we initialized the *et* at 70 ms. However, we updated it as a first order autoregressive variable, depending on the time difference between when the robot was supposed to hit and when it hit the drum.

For example, suppose a robot needs to hit the drum at time $t_{(n+1)}$ predicted by our algorithms (in the *AlgoNode*). After generating a prediction, the *AlgoNode* sends this timing to the *ArduinoNode* to execute the hit at $t_{(n+1)}$. The *ArduinoNode* starts the drumming action at time $t_{(n+1)} - et$. Due to physical noise, the robotic arm may hit the drum a little earlier or later than $t_{(n+1)}$. Suppose that the robotic arm hits the drum at time $t_{hit}$. We update *et* for the next hit event based on the difference between $t_{(n+1)}$ and

$t_{hit}$, and a movement alignment factor $f$. We describe this process in Equation 10.16. For our robot, we set $f = 0.2$.

$$et_{(i+1)} = et_i + f * (t_{(n+1)} - t_{hit}) \tag{10.16}$$

### 10.2.3 Methodology

We performed a 3 x 3, within-subjects study to explore our research goals. We used three different algorithms for the robot to drum with people in three different rhythmic conditions: 1) adaptation only, 2) anticipation only, and 3) combined dyadic algorithms. (See Section 10.2.3). Also, we employed three tempo-changing rhythmic signals, which we describe in detail in Section 10.2.3.

#### Experimental Conditions

Person 1 was instructed to simply observe the first four beats of the rhythm and to start drumming from the fifth beat. These four beats served to help the person to perceive the tempo.

During each session, one person acted as *Person 1*, and the other person acted as *Person 2* for all nine experimental conditions. These sessions were then repeated once more, with each of the participants swapping roles. The experimental conditions were counter-balanced. The whole set of experiments per group took about 40-50 minutes in total.

During the experiments, we used a phase correction parameter $\alpha = 0.3$, a period correction parameter $\beta = 0.5$, an anticipation parameter $k = 3$, a prediction-tracking ratio $m = 0.8$, and an anticipatory correction gain $\gamma = 0.4$. The $\alpha$ and $\beta$ values were selected based on the recommendation by Van Der Steen el al. [216]. The other parameter values were selected from pilot experiments.

**Rhythms**

We used three rhythmic tempos for this experiment. Each rhythm had 76 beats in total.

The first rhythm was a uniform tempo. The signal had a tempo of 80 BPM (beats per minute) and lasted for 76 beats. *Person 1* skipped the first four beats and drummed for the next 72 beats. Each metronome beat lasted for 5 ms.

The second rhythm started with a tempo of 80 BPM for first 20 beats. It slowed down gradually (*ritardando*) starting from the 21st beat for next 24 beats to reach 60 BPM. After reaching 60 BPM, the tempo sped up (*accelerando*) from the 45th beat for 24 beats to reach 80 BPM again. The rhythm then ended in 80 BPM for the last 8 beats.

The third rhythm also had a similar tempo change like rhythm 2. It started with a tempo of 80 BPM. However, this rhythm had two *ritardando* and *accelerando* phases, and each phase lasted for 12 beats. During the ritardando and accelerando phases, the tempo changed from 80 BPM to 60 BPM and returned to 80 BPM again. Figure 10.6 shows the tempo changes.

We used a sigmoid function to generate the *ritardando* and *accelerando* phases, similar to one used by Van Der Steen et al. [216]. Equations 10.17 and 10.18 define the sigmoid function. Here, *IOI* represents inter-onset interval. For example, during the *ritardando* step, the $IOI_{start}$ was 750 ms (80 BPM) and $IOI_{goal}$ was 1000 ms (60 BPM). $N_{steps}$ represents the number of steps it will take to complete the tempo change. For our single tempo change phase, it was set to 24.

$$IOI_i = IOI_{goal} + (IOI_{start} - IOI_{goal}) * fr \tag{10.17}$$

$$fr = \frac{1}{2}[1 + cos\frac{i-1}{N_{steps}-1}\pi] \tag{10.18}$$

### Procedure

After giving informed consent, both participants were instructed about the experimental procedure. The participants then had time to practice the drumming session as a group. During this practice session, the robot did not drum with them.

The participants practice for two rounds, first as $P1$ and then as $P2$. We used a uniform tempo condition (60 BPM) for these rounds, which was different than the uniform tempo that we used for the experimental conditions (80 BPM).

### Participants

For our main experiment, we recruited a total of eight groups (16 participants in total, two persons per group). 4 participants were women, 12 were men. Their average age was 24 years (s.d. = 2.38 years), and the majority were undergraduate and graduate students. Only 3 participants had prior drumming experience of two years, and 13 did not have any drumming experience. Participants were recruited via word-of-mouth. Upon scheduling a time slot, participants were randomly assigned to join a group with another participant. Each participant was compensated with a 5 gift card for their time.

## 10.3 Data Analysis and Results

To compare the performance of the algorithms, we first measured how closely the robot drummed to *Person 1*. We also measured how closely *Person 2* drummed to *Person 1*. Then we performed a non-inferiority trial on these two measures. This trial enables us to investigate whether the robot's performance was at least as good as the human performance.

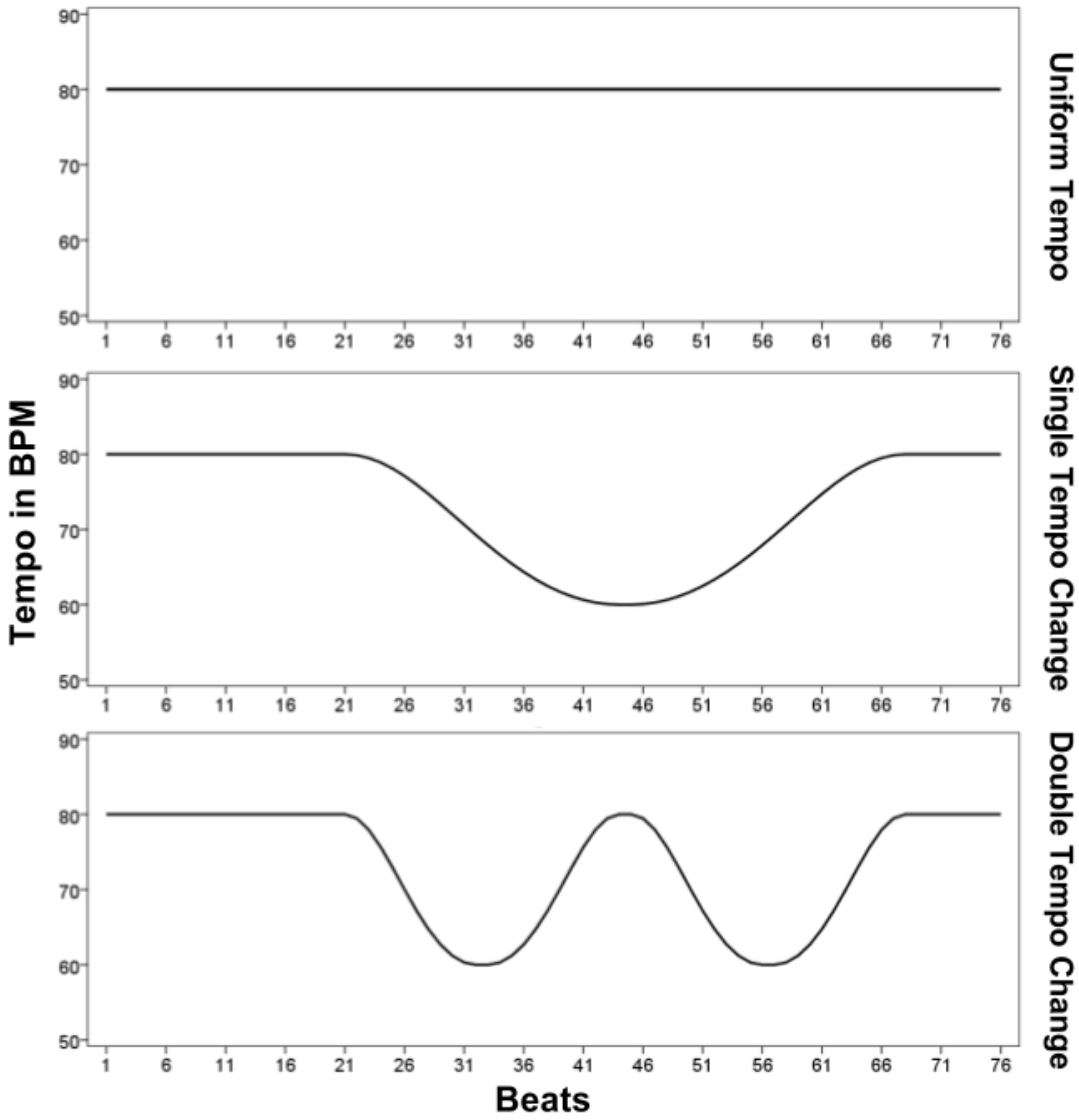


**Figure 10.6**: Three tempo conditions for our experiments. A) Uniform tempo condition, B) Single tempo change condition, and C) Double tempo change condition.

### 10.3.1 Data Processing

During our experiments, we logged all the timing of events for both participants and the robot. We first time-aligned the data. Next, we only took the events that occurred starting from the 18th beat of the metronome signal. Recall that *Person 1* observed the first four beats of the metronome signal, and then started drumming on the fifth beat. The robot observed *Person 1* for next eight beats, and joined in the drumming on 13th beat. For next five beats, the robot drummed by using only anticipation' algorithm. The robot began performing using different algorithms starting from the 18th beat of the metronome signal based on the experimental condition. Thus, we used beats 18 through 76 (a total of 59 beats) for our calculation.

We excluded all data that did not capture drum events from the robot or the participants from our calculation. Thus we excluded about 5.13

We then calculated the timing of all drumming events performed by *Person 1 (P1)*, *Person 2 (P2)* and the *robot (R)*. For example, for a single drumming event $e$, we calculated the timing of the event performed by *P1* and *P2*, and the robot respectively as $t(P1,e)$, $t(P2,e)$, and $t(R,2)$. Then, we measured the absolute difference between the timing of each event performed by $P1$ and $P2$, and $P1$ and $R$. Thus, we computed $abs(t(P1,e)-t(P2,e))$ and $abs(t(P1,e)-t(R,e))$.

After measuring the offset of each event for $P2$ and $R$ from $P1$, we averaged it over each session. For example, if a session has a total of a $E$ number of drumming events, where $e \in E$, then we measured the mean difference of timings ($dt$) as:

$$dt(P1,P2) = \frac{\sum_{e=1}^{E} abs(t(P1,e)-t(P2,e))}{E} \tag{10.19}$$

$$dt(P1,R) = \frac{\sum_{e=1}^{E} abs(t(P1,e)-t(R,e))}{E} \tag{10.20}$$

This data ($dt(P1,P2)$ and $dt(P1,R)$) provide us the information about how closely $P2$ and $R$ matched their rhythms to $P1$. We then performed a non-inferiority trial over these data, which I describe in the next Section.

### 10.3.2 Analysis Method

We performed a non-inferiority trial, which enabled us to investigate whether the robot's performance was at least as good as the human performance. The non-inferiority trial is a widely used analysis technique, particularly used in the healthcare domain. In those domains, non-inferiority trials are usually used to test whether a new treatment is not unacceptably worse than an active treatment that is already in use [190, 160, 56, 200].

Researchers in the robotics community are usually interested in testing whether a new method is better than an existing method. These are called superiority trials.

In contrast, a non-inferiority trial tries to show that a new method is not inferior to an existing method [190, 200]. In other words, a new method is not unacceptably worse than an existing method. The new method is used as the experimental condition, and the existing method is used a the control condition.

A non-inferiority trial allows a limit on the amount an experimental condition is permitted to be inferior to a control condition [190]. This also guarantees a similar benefit and efficacy from the experimental condition.

The acceptable amount for which an experimental condition may be worse than a control condition is called a non-inferiority margin. It is usually denoted by $\delta$. If an experimental condition is within the non-inferiority margin of the control condition, the experimental condition is considered as non-inferior to the control condition and is considered as beneficial as the control condition [190, 56].

This margin may represent a threshold for the difference in effects of the experimental condition and the control condition [190]. It also may represent a threshold for

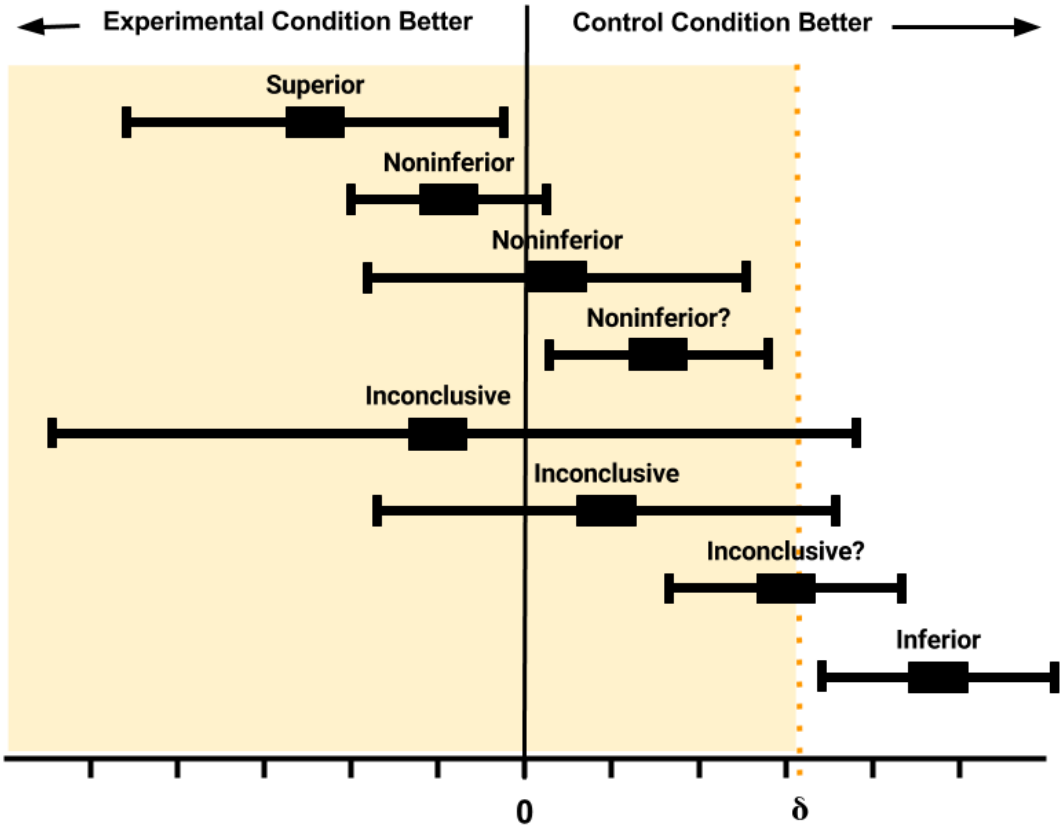


**Figure 10.7**: Different conclusions for non-inferiority trials. (Adapted from [160])

the relative size of the effects. However, in practice, this margin is usually subjective. Typically, this margin is chosen based on a review of the effect of the control condition, so that the experimental condition would be regarded as beneficial and effective as the control condition.

For a non-inferiority trial, the difference between the effect of the experimental condition and the control condition is calculated. To calculate this, the effect of the control condition is subtracted from the effect of the experimental condition. Then from the distribution of these values (i.e., 2-sided 95% confidence intervals), the non-inferiority is tested. Figure 10.7 shows various conditions for determining non-inferiority [160].

### 10.3.3 Results

In our case, we considered the time difference between $P1$ and $P2$ as our control condition. We want to test whether the robot performed the drumming actions at least as close as the human ($P2$). Thus, the time difference between $P1$ and $R$ is our experimental condition.

For our non-inferiority trials, we subtracted the effect (in our case, the closeness of drumming times to $P1$) of the control condition from the experimental condition. Thus, we calculated $dt(P1,R) - dt(P1,P2)$ for each condition. These values are represented in milliseconds (ms). The negative values represent that the robot performed closer to $P1$ in times than $P2$. On the other hand, a positive value represents that $P2$ performed closer to $P1$ in times than the robot.

We show the results on the non-inferiority test in Table 10.1. Here, we present $dt(P1,R) - dt(P1,P2)$ for all our experimental conditions. We also present the 95% confidence interval of $dt(P1,R) - dt(P1,P2)$ in last two columns of this table.

To perform a non-inferiority trial, we needed to choose a non-inferiority margin ($\delta$). We choose $\delta = 70$ ms. We choose this value for $\delta$, as this is an approximate time that our robot takes to hit the drum pad after receiving a command. If we choose a value larger than this, then a reactive robot (which can react and can start drumming after listening to a person hit) would be better than our algorithms. On the other hand, being closer than this margin indicates that the algorithms are better than by chance (reactive controls). Therefore, this the margin ($\delta = 70$ ms) was chosen.

In Figure 10.8, we present the the 95% confidence interval of $dt(P1,R) - dt(P1,P2)$. Here, we present this value for all 9 conditions of our experiments (3 robot algorithms X 3 rhythms). We marked the 0-th line and the 70-th line (our $\delta$).

Figure 10.8 and Table 10.1 indicate that for uniform tempo condition, the anticipation algorithm performed non-inferior to the control condition (95 % confidence interval,

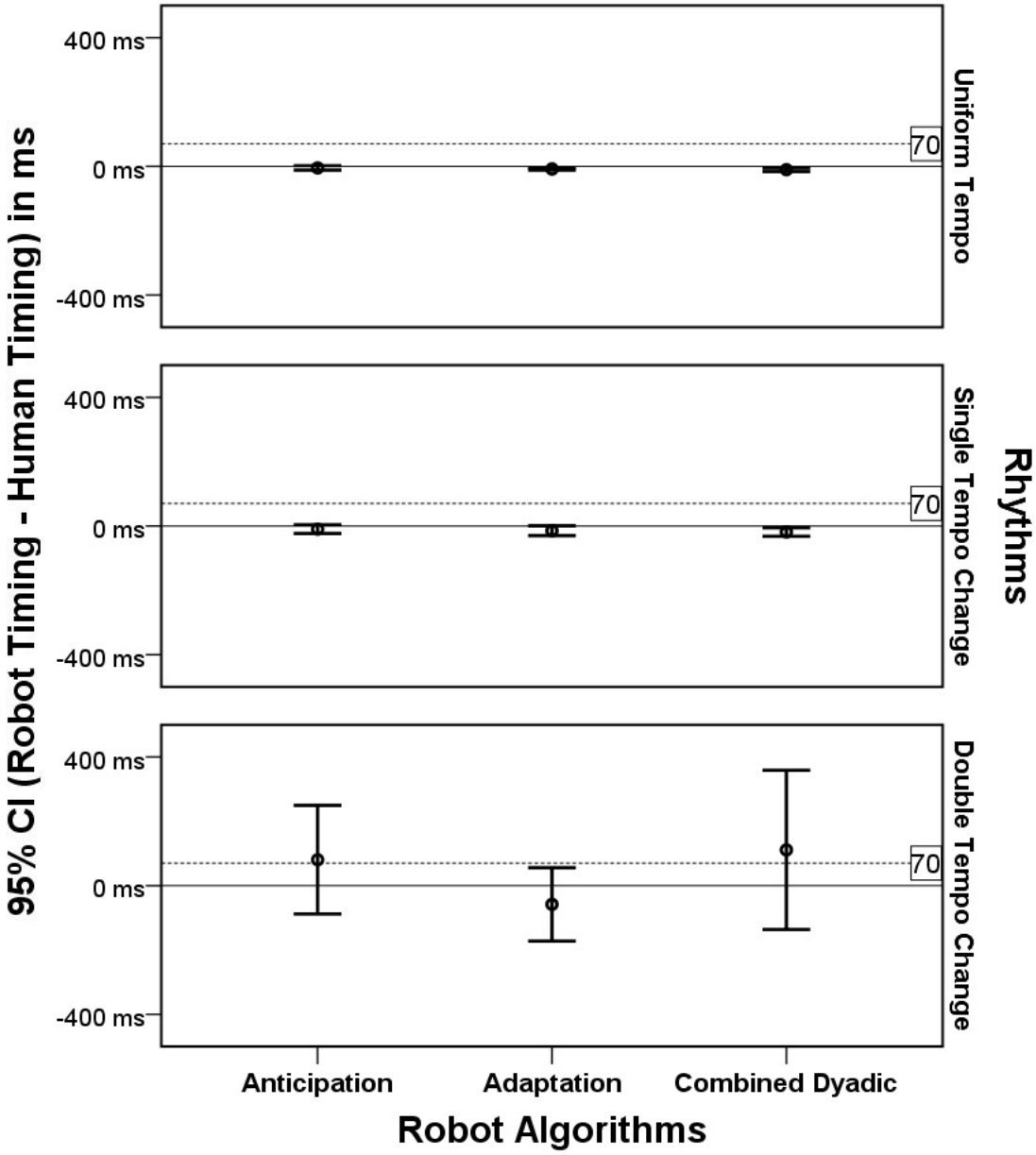


**Figure 10.8**: The results of the non-inferiority trials. The algorithms are presented along $x$-axis, and three tempo conditions are presented in separate boxes along $y$-axis. 95% confidence interval of $dt(P1,R) - dt(P1,P2)$ in ms are presented along $y$-axis. The dashed line represents the non-inferiority margin ($\delta = 70ms$).

**Table 10.1**: Presents $dt(P1,R) - dt(P1,P2)$ for all nine experimental conditions.

| | | dt(P1 , R) - dt(P1 , P2) | | | |
|---|---|---|---|---|---|
| **Robot Algorithms** | **Rhythms** | **Mean** | **s.d.** | **95% CI** | |
| | | | | **Lower** | **Upper** |
| Anticipation | Uniform Tempo | -4.82 | 12.92 | -11.70 | 2.07 |
| Anticipation | Single Tempo | -9.74 | 25.59 | -23.38 | 3.89 |
| Anticipation | Double Tempo | 80.62 | 316.78 | -88.17 | 249.43 |
| Adaptation | Uniform Tempo | -8.09 | 6.86 | -11.75 | -4.43 |
| Adaptation | Single Tempo | -14.47 | 28.53 | -29.68 | 0.73 |
| Adaptation | Double Tempo | -58.04 | 213.78 | -171.96 | 55.87 |
| Combined Dyadic | Uniform Tempo | -10.26 | 10.79 | -16.01 | -4.51 |
| Combined Dyadic | Single Tempo | -18.47 | 25.18 | -31.89 | -5.06 |
| Combined Dyadic | Double Tempo | 111.26 | 464.64 | -136.33 | 358.85 |

$-11.70$ to 2.07), the adaptation algorithm performed non-inferior and also superior to the control condition (95 % confidence interval, $-11.75$ to $-4.43$), and the combined dyadic algorithm performed non-inferior and also superior to the control condition (95 % confidence interval, $-16.01$ to $-4.51$).

For the single tempo-changing condition, the anticipation algorithm performed non-inferior to the control condition (95 % confidence interval, $-23.37$ to 3.88), the adaptation algorithm performed non-inferior to the control condition (95 % confidence interval, $-29.68$ to 0.73), and the combined dyadic algorithm performed non-inferior and also superior to the control condition (95 % confidence interval, $-31.89$ to $-5.07$).

For the double tempo-changing condition, the performance of the anticipation and the algorithm were inconclusive compared to the control condition (Anticipation: 95 % confidence interval, $-88.17$ to 249.43, and Combined dyadic: 95 % confidence interval, $-136.33$ to 358.85). However, the adaptation algorithm performed non-inferior

**Table 10.2**: Presents the summary of the findings of the non-inferiority trials. *Non-inferior* means that the robot was as good as an average human drummer, *Superior* means that the robot was better than an average human drummer, and *Inconclusive* means that we are unable to determine the difference between the two.

| **Robot Algorithms** | **Rhythms** | | |
|---|---|---|---|
| | **Uniform Tempo** | **Single Tempo Change** | **Double Tempo Change** |
| **Anticipation** | Non-inferior | Non-inferior | Inconclusive |
| **Adaptation** | Superior | Non-inferior | Non-inferior |
| **Combined Dyadic** | Superior | Superior | Inconclusive |

to the control condition (95 % confidence interval, $-171.97$ to 55.87).

We summarized these findings in Table 10.2.

## 10.4 Discussion

The results suggest that the adaptation algorithm was non-inferior to the control condition in all three rhythmic conditions, even was superior in uniform tempo condition. Similarly, the combined dyadic algorithm was superior to the control condition (compared to human timing) in the uniform tempo and the single tempo-changing conditions. The anticipation algorithm was non-inferior to the control condition in the uniform tempo and the single tempo-changing condition.

We cannot conclude that the combined dyadic algorithm and the anticipation algorithm were non-inferior to the control condition in the double tempo-changing condition. This is because the upper limits of the 95% CI of $dt(P1,R) - dt(P1,P2)$ are greater than $\delta$ (70) and the lower limits are less than 0.

Overall, the results suggest that the adaptation algorithm alone enables a robot to achieve a human-level performance when that person drums synchronously with another

person only by relying on auditory signals. By combining anticipatory knowledge (anticipation algorithm), along with an adaptation process, a robot can be even better than people for the uniform and single tempo-changing conditions.

Although the anticipation algorithm was non-inferior to human performers during the uniform and single tempo-changing conditions, we cannot draw the same conclusion for the double tempo-changing condition. We believe that this may be because there were faster tempo changes during the double tempo change condition than in the other two conditions.

For example, the *ritardando* and the *accelerando* phases lasted for 24 beats during the single tempo-changing condition to change the tempo from 80 BPM to 60 BPM. In other words, the inter-tone-interval changed from 750 ms to 1000 ms in 24 beats. These phases lasted only for 12 beats in the double tempo-changing condition. Thus, the inter-tone-interval changed from 750 ms to 1000 ms in 12 beats, which is faster than the single tempo-changing condition.

One possible reason why some effects were inconclusive in the double-tempo changing condition may be due to the fact that the anticipation algorithm only uses linear extrapolation to calculate events. This may be insufficient to predict future event timings.

The main contribution of this work is the development of models for robots to better understand tempo-changing behavior, and to evaluate their effects while coordinating with people. The results suggest that a robot can adapt to rhythmic changes at least as well as an average person, and even better in many cases. This work allows us to build adaptive algorithms for robots to better understand group coordination, even when the team dynamics changes.

In the future, we plan to explore the effect of these adaptation and anticipation algorithms in a group context (described in Section 10.1.4). This exploration will give us an opportunity to investigate many dimensions of the problem that will need attention

during future robot algorithm development and human-robot interaction design.

Moreover, as people will encounter various kinds of robots with different capabilities while working in teams in the future, it is necessary to investigate the effects of the presence of multiple robots with varying levels of adaptability in teams. We further want to explore the effect of these algorithms while coordinating in the context of multi-human and multi-robot teams, which will give us a necessary understanding of the underlying human-robot team dynamics in such scenarios.

## 10.5 Chapter summary

In this Chapter, I described the design and implementation of four algorithms for robots to understand tempo changes in rhythms. The design of these algorithms was inspired by the neuroscience literature. I validated the efficacy of these algorithms by applying to a robot to drum synchronously with a person, even when the tempo of the drumming changed. These algorithms were found to be effective, and have the potential to be implemented on robots to interact with people in teams.

## 10.6 Acknowledgements

I thank Auriel Washburn for her help to design the experiment, and Peter Keller for sharing early implementations of ADAM. I thank Darren Chan and Maryam Moosaei for their help in building the robot and testing various sensors, Angelique Taylor and Bryce Woodworth for their valuable advice on my study design, and Ji-eun Park for helping design the figures used in this Chapter.

This chapter contains material from “Tempo Adaptation and Anticipation Methods for Human-Robot Teams,” by T. Iqbal., M. Moosaei, and L. D. Riek, which appears

in Proceedings of the Robotics: Science and Systems Conference, Planning for HRI: Shared Autonomy and Collab. Robotics Work, 2016 [77], and "Can robots synchronize with humans in tempo changing environments?" by T. Iqbal. and L. D. Riek, which appears in Proceedings of the Conference of Joint Action Meeting, 2017 [86]. I was the primary investigator and author of these papers.

# Chapter 11

# Conclusion

This chapter discusses the main contributions of my work for the human-robot interaction and robotics communities in general. It then briefly introduces some work that I would like to conduct in the future. The chapter concludes with some open questions that I discovered over the course of my research which need to be addressed to build fluent human-robot teams.

## 11.1 Contributions

### 11.1.1 Developed a new non-linear method to measure the synchronization of a group

Interaction in a team is an important aspect of human social behavior, and is an active area of research in many fields including robotics, psychology, cognitive science, kinesiology, dance, and music. If a robot could make better sense of how humans interact among themselves in a team, its interactions with people would reach a higher level of coordination, resulting in a fluent meshing of actions.

Understanding human team dynamics is not trivial. One of the main challenges is

the unpredictability of human behavior. Another major challenge is accurately detecting human activities "in the wild", e.g., in ecologically realistic settings. However, if a robot has some ability to model team dynamics, then it can anticipate future actions in a team, and adapt to those actions in order to be an effective teammate.

To enable a robot to understand team dynamics, I developed a method to automatically detect the degree of synchronization in a team (discussed in Chapter 4). This method takes multiple types of discrete, task-level events into consideration. It is capable of modeling synchronous and asynchronous behavior of the group, and is well-matched to the group members' perception about synchrony. Our method is more accurate in estimating synchrony than other methods from the literature that depend on a single event type [82].

I validated our new method across a series of human-human and human-robot teaming experiments, which included automatic detection of synchrony in human teams (see Chapter 5), synchronization synthesis in robots (see Chapter 6), and development of a new event anticipation algorithm for robots (see Chapters 7 and 8).

### 11.1.2 Validated the new method by applying it to a human-human team

For human-human teams, II validated the method in three ways (described in detail in Chapter 5). First, I applied the method to analyze the synchrony during a multi event-based rhythmic game. This involved a group game where each player performed a periodic psychomotor activity within the group. Each player's movements influenced the actions of all other group members. Our method was able to model both synchronous and asynchronous activities and was well-matched to the players' perceptions of synchrony [82, 80, 81].

Second, I compared our results to another method from the literature that uses

single event types. The results suggest that our multiple event-based method is more accurate in estimating synchrony than a single event-based method.

Third, I compared our results to the cross-recurrence quantification analysis (CRQA) method, which is widely used in the literature to measure synchronization in a system. The results again suggested that our multiple event-based method is more accurate in estimating synchrony in human teams.

### 11.1.3 Validated the new method by applying to a human-robot team

I further validated our method by applying it to a human-robot team (described in Chapter 6). In this work, we used multiple robots to automatically measure the synchronous joint action of a group while both the robots and co-present humans were in motion. During the experiments, two people marched either synchronously or asynchronously, while being followed by two autonomous mobile robots. The results suggested that our method accurately identifies synchronous and asynchronous motion in the team [76, 75].

### 11.1.4 Designed a new approach for robots to anticipate and synthesize coordinated actions

I designed a new approach to enable robots to perceive human group motion in real-time, anticipate future actions, and synthesize their own motion accordingly. I explored this within the context of joint action, where humans and robots moved together synchronously [79, 78, 166, 83].

Our approach (the synchronization index-based anticipation (SIA) algorithm) takes group coordination dynamics into account while anticipating the timing of a future

event. I developed another motion anticipation algorithm (event cluster-based anticipation (ECA) method) to serve as a reasonable comparison to our developed method, which did not take group's internal dynamics into account.

Next, I validated these methods (SIA and ECA) within a human-robot interaction scenario, where an autonomous mobile robot observed a team of human dancers, and then successfully and contingently coordinated its movements to "join the dance." It was able to do that without having any prior knowledge of the dance sequences.

The results suggested that the SIA method performed better both in terms of more closely synchronizing the robot's motion to the team, and also exhibiting more contingent and fluent motion than the ECA method.

These findings suggest that robots perform better when they have an understanding of team dynamics than when they do not.

### 11.1.5 Investigated how the presence and behavior of robots affect group coordination in multi-human multi-robot teams

As robots enter human environments, they will be expected to collaborate and coordinate their actions with people. In many of these situations, multiple robots will be expected to collaborate with multiple people in teams.

How robots engage in the coordination process can influence the dynamics of the team, particularly in multi-human, multi-robot situations. I investigated how the presence of robots affects group coordination when both the anticipation algorithms they employed and the number of robots varied (single robot or multi-robot) (see Chapter 9) [87, 84, 85].

The results suggested that group coordination was significantly degraded when a robot joined a human-only group, and was further degraded when a second robot joined the team and employed a different anticipation algorithm from the other robot. These

findings suggest that heterogeneous behavior of robots in a multi-human group can play a major role in how group coordination dynamics change.

### 11.1.6 Created human-inspired adaptation and anticipation models for robots

Beyond the rhythmic synchronous team interactions scenarios, people also effectively time their actions to coordinate with others during everyday activities, even in a constantly changing environment. However, robots still do not know how to take advantage of such concepts.

I designed and implemented algorithms for robots to coordinate with the other members of a team in tempo-changing environments, inspired by the neuroscience literature (see Chapter 10). These algorithms leveraged a human-like understanding of temporal anticipation and adaptation during the coordination process [77, 86].

I validated the algorithms by applying them in a human-robot synchronous drumming scenario. The results suggested that an adaptation algorithm alone enabled a robot to achieve human-level performance in a synchronous drumming scenario. By combining an anticipatory knowledge along with an adaptation process, a robot can perform as good as, or sometimes better than, people during several tempo-changing situations.

### 11.1.7 Developed a time-synchronous multi-modal data capture system

I identified many of the aspects and challenges of building a time-synchronous data capture system, which is an essential element to investigate group interactions precisely. From these, I designed and implemented a time-synchronous, multi-modal data

capture system based on a client-server architecture to detect high-level human activities (described in Chapter 3) [74].

I deployed this system to capture data synchronously with two Kinect sensors during a dance-off scenario. Based on the captured data, a motion analysis algorithm made decisions about human movements. From these decisions, the system passed movement commands to a mobile robot.

## 11.2 Future Work

### 11.2.1 Coordinating with groups in tempo-changing environments

In chapter 10, we introduced four different modules for robots to understand and respond to dynamic tempo changes in their surroundings. We validated three of the developed modules in a human-robot teaming context and left the fourth module (Combined Module: Group) for our future exploration.

As robots are becoming increasingly common in teams, they will encounter people and need to adapt to their actions. In group settings, a robot may find that different people are working at different paces. To coordinate with the team, the robot needs to understand all those various pacing of actions and build a strategy for itself.

In my future work, I want to validate the anticipation and the adaptation algorithms that we developed in Section 10.1.4 in a human-robot teaming context. I want to explore how effectively a robot can understand various rhythms and can make appropriate decisions to incorporate itself into a group.

I also want to explore how multiple robots can take advantage of these algorithms to interact with multiple people in a team. This exploration is important, as understanding how coordination dynamics change in these scenarios may be an important metric for us to design robots' strategy for teams.

### 11.2.2 Understanding coordination in non-rhythmic tasks

Beyond rhythmic coordination, a robot may encounter many group tasks, where the task might not be rhythmic and cyclical. However, there may exist some structural similarities how the group interacts to achieve the goal in those tasks.

In the future, I plan to explore coordination beyond synchronous joint action. Although my research up to this point explores a subset of entire coordination problem domain, our algorithms have potential to be extended in other scenarios. I want to extend our algorithms for other coordination tasks, which may not be synchronous but have some structural similarities.

### 11.2.3 Leveraging non-verbal cues during human-robot interaction

In human-human communication, we convey a significant portion of information non-verbally, such as through our hand gestures, head movements, postures, etc. Thus, if researchers only focus on solving problems in natural language processing, a significant portion of the message is lost [176].

To address this challenge, I plan to explore how a robot can incorporate an understanding of non-verbal human cues along with an understanding of verbal messages. I also plan to explore how to fuse this understanding of non-verbal cues with verbal messages to understand the conversational context more accurately, so that we can use that knowledge to synthesize the gestures of a robot more appropriately.

### 11.2.4 Anticipatory action selection for assistive robots

Robots can help support older adults with mobility, manipulation, and cognition. For example, older adults may need help to retrieve medication, prepare meals, locomote, or perform household tasks. If a robot can anticipate the needs and intentions of a

person in advance, then it can perform its tasks efficiently and will be more helpful and functional.

I plan to develop robotic systems which will be able to understand the intentions of older adults from their verbal messages and non-verbal cues in real-time. Along with this understanding, these systems will incorporate contextual knowledge to anticipate the future intentions. Another advantage to such an approach is that it has the potential to be adaptive to the unique differences between people, especially since their situations change as they age.

## 11.3 Open Questions

### 11.3.1 Will it be possible for a robot to accurately understand its role in a dynamically changing group?

If a robot has some ability to understand its role in a dynamically changing group, its interaction will be more fluent with the team. However, understanding a role in a team is non-trivial.

In many human-human team situations, team members are explicitly assigned to various roles [153]. On the other hand, in many other team scenarios, various roles emerges over time across the team members to achieve a common goal [105]. Often these assigned roles change dynamically based on the necessity of the task.

For example, a person who begins to lead a team to move a table may follow another teammate's lead later during the moving process. How people coordinate and cooperate amongst themselves in these situations is an important indicator for robots to understand the role distributions in the group.

In human-robot interaction scenarios, various predefined role distribution models

are used based on a task. However, these well-defined role distributions are rarely seen in real-world situations. Distributed roles change dynamically in many situations. If the roles for an interaction are not predefined, the robot needs to make predictions about the role of co-present people, in order to infer its role in the group. A dynamic understanding of role distributions in a human-robot team can enable a robot to understand team dynamics more appropriately, which can lead to fluent interaction in the group.

### 11.3.2 Can we build robots for long-term interaction?

For decades, people have been using robots to solve problems in industrial settings (such as automobile manufacturing), in which the robots performed repetitive tasks for long periods of time. Recently, robot usage has begun to expand into human social environments. In many of these settings, robots will be expected to provide direct assistance to people for many years at a time, for example by giving support to older adults in their homes.

While working in such social environments, robots will need to fluently interact with people, learn from their interactions, adapt to people's individualized needs, and modify their behaviors accordingly over time. Extensive research needs to be conducted in order to enable future robots to perform such long-term social interactions, including work on human behavior modeling, learning from demonstration, long-term shared autonomy, understanding coordination dynamics, and communication strategies, to name a few.

### 11.3.3 Does a robot need better human action detection mechanisms to act fluently with people?

One of the main challenges to detecting human actions is the unpredictability of human behavior. Sometimes it can be difficult for a robot to perceive and understand the different types of events involved in these activities to make effective decisions due to sensor occlusion, sensor fusion error, unanticipated motion, narrow field of view, cluttered backgrounds, etc. [180, 176, 23, 22, 168].

One approach to address the challenge of human action detection is to use classification algorithms to detect actions from video data. However, this approach has major challenges, including intra vs inter-class variations between action classes, temporal variations of actions, and obtaining and labeling training data [161]. Moreover, using a classifier for action detection has several computational bottlenecks, including generalizability, abnormality detection, and classifier training [168]. Furthermore, in most action recognition cases, researchers usually assume that camera positions are static, which is not the case for mobile robots [22].

If a robot can detect human actions accurately, its ability to coordinate with people will be improved. In the future, robots need better hardware and algorithms to address these practical challenges to be effective in human environments.

## 11.4 Closing Remarks

My Ph.D. research explores coordination dynamics in human-robot teams. I explore complex real-world problems pertaining to building intelligent robots capable of working with people in real-world scenarios.

The primary contribution of my research is the introduction of new algorithms that enable robots to understand coordination dynamics, as well as procedures to synthesize

appropriate actions in human-robot teams. As robots are becoming increasingly common in human social environments, it is becoming even more important for them to understand team dynamics, to anticipate future actions, and to take appropriate adaptive actions. My Ph.D. work focuses on addressing some of the exciting and important research questions regarding enabling robots to work fluently with people in teams.

I hope that my work will help to build adaptable, efficient, and functional robots for multi-human, multi-robot teams.

# Bibliography


[1] H. Admoni, A. Dragan, S. Srinivasa, and B. Scassellati. Deliberate Delays During Robot-to-Human Handovers Improve Compliance With Gaze Communication. *Int. Conf. Human-Robot Interact.*, 2014.

[2] S. Afzal and P. Robinson. Natural affect data: Collection and annotation. In *New Perspectives on Affect and Learning Technologies*. 2011.

[3] J. Aggarwal and M. Ryoo. Human activity analysis. *ACM Comput. Surv.*, 2011.

[4] C. Alexopoulos, D. Goldsman, and J. R. Wilson. Overlapping batch means: something more for nothing? In *Proc. of the Winter Simulation Conference*, 2011.

[5] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[6] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters. Interaction primitives for human-robot cooperation tasks. In *IEEE Conf. on Robotics and Automation*, 2014.

[7] B. Argall, S. Chernova, and M. Veloso. A Survey of Robots Learning from Demonstration. *Robotics and Autonomous Systems*, 2009.

[8] J. Bandouch, O. C. Jenkins, and M. Beetz. A Self-Training Approach for Visual Tracking and Recognition of Complex Human Activity Patterns. *Int. J. Comput. Vis.*, 2012.

[9] E. I. Barakova and T. Lourens. Expressing and interpreting emotional movements in social games with robots. *Pers. Ubiquitous Comput.*, 14(5), 2010.

[10] J. Bluemink, R. Hämäläinen, T. Manninen, and S. Järvelä. Group-level analysis on multiplayer game collaboration: how do the individuals shape the group interaction? *Interact Learn Envir*, 2010.

[11] A. Bobick and V. Krüger. On human action. In *Visual Analysis of Humans*. 2011.

[12] S. M. Boker and J. L. Rotondo. Symmetry building and symmetry breaking in synchronized movement. *Adv Consc Res*, 2002.

[13] J. M. Bradshaw, P. Feltovich, M. Johnson, M. Breedy, L. Bunch, T. Eskridge, H. Jung, J. Lott, A. Uszok, and J. van Diggelen. From tools to teammates: Joint activity in human-agent-robot teams. In *Human Centered Design*. 2009.

[14] O. Brdiczka, J. Maisonnasse, and P. Reignier. Automatic detection of interaction groups. In *Proc. Int. Conf. Multimodal Interfaces*, 2005.

[15] M. Bretan and G. Weinberg. A survey of robotic musicianship. *Communications of the ACM*, 59(5):100–109, 2016.

[16] T. Brys, A. Harutyunyan, V. U. Brussel, and M. E. Taylor. Reinforcement Learning from Demonstration through Shaping. *Proc. Twenty-Fourth Int. Jt. Conf. Artif. Intell.*, 2015.

[17] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru. An adaptable system for rgb-d based human body detection and pose estimation. *J Vis Commun Image R*, 2013.

[18] M. Cakmak, S. S. Srinivasa, M. K. Lee, S. Kiesler, and J. Forlizzi. Using spatial and temporal contrast for fluent robot-human hand-overs. In *Proc. of ACM/IEEE HRI*, 2011.

[19] N. Campbell. Multimodal Processing of Discourse Information; The Effect of Synchrony. *Int. Symp. Univers. Commun.*, 2008.

[20] T. Carlson and Y. Demiris. Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload. *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, 42(3):876–88, jun 2012.

[21] G. Castellano, I. Leite, A. Pereira, C. Martinho, A. Paiva, and P. W. Mcowan. Multimodal affect modeling and recognition for empathic robot companions. *International Journal of Humanoid Robotics*, 2013.

[22] D. Chan and L. D. Riek. Object proposal algorithms in the wild: Are they generalizable to robot perception? In *In review*, 2017.

[23] D. Chan, A. Taylor, and L. D. Riek. Faster robot perception using salient depth perception. In *In review*, 2017.

[24] X. Chang, W.-s. Zheng, and J. Zhang. Learning Person  Person Interaction in. 24(6):1905–1918, 2015.

[25] C. Chao and A. Thomaz. Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets. *J. Human-Robot Interact.*, 2012.

[26] M. J. Choi, A. Torralba, and A. S. Willsky. Context Models and Out-of-context Objects. *Pattern Recognition Letters*, 2011.

[27] W. Choi, C. Pantofaru, and S. Savarese. A general framework for tracking multiple people from a moving camera. *IEEE T Pattern Anal*, 2013.

[28] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Computer Vision ECCV 2012*, 2012.

[29] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2011.

[30] M. Clayton, R. Sager, and U. Will. In time with the music: the concept of entrainment and its significance for ethnomusicology. *European Meetings in Ethnomusicology*, 2005.

[31] M. I. Coco and R. Dale. Cross-recurrence quantification analysis of categorical and continuous time series: an r package. *Frontiers in psychology*, 2014.

[32] J. W. Crandall, M. A. Goodrich, D. R. Olsen Jr, and C. W. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(4), 2005.

[33] F. Cummins. Rhythm as entrainment: The case of synchronous speech. *J Phonetics*, 2009.

[34] R. Dale and M. J. Spivey. Unraveling the dyad: Using recurrence analysis to explore patterns of syntactic coordination between children and caregivers in conversation. *Lang Learn*, 2006.

[35] R. Dale, A. S. Warlaumont, and D. C. Richardson. Nominal cross recurrence as a generalized lag sequential analysis for behavioral streams. *Int J Bifurcat Chaos*, 2011.

[36] K. Dautenhahn, S. Woods, C. Kaouri, M. Walters, K. L. Koay, and I. Werry. What is a robot companion - friend, assistant or butler? In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS).*, 2005.

[37] C. De Looze, S. Scherer, B. Vaughan, and N. Campbell. Investigating automatic measurements of prosodic accommodation and its dynamics in social interaction. *Speech Commun*, 2014.

[38] E. Delaherche, M. Chetouani, A. Mahdhaoui, C. Saint-Georges, S. Viaux, and D. Cohen. Interpersonal Synchrony: A Survey of Evaluation Methods across Disciplines. *IEEE T on Affective Computing*, 2012.

[39] S. D'Mello and J. Kory. Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. In *Proc. of ACM ICMI*, 2012.

[40] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, U. T. Austin, U. Lowell, and U. C. Berkeley. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *Computer Vision and Pattern Recognition*, 2015.

[41] A. D. Dragan, K. C. Lee, and S. S. Srinivasa. Legibility and predictability of robot motion. In *Proc. of ACM/IEEE HRI*. IEEE, 2013.

[42] A. D. Dragan, A. L. Thomaz, and S. S. Srinivasa. Collaborative manipulation: new challenges for robotics and hri. In *Proc. of ACM/IEEE HRI*, 2013.

[43] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhys Lett*, 1987.

[44] J. Fasola and M. Mataric. A Socially Assistive Robot Exercise Coach for the Elderly. *J. Human-Robot Interact.*, 2013.

[45] T. Flash and B. Hochner. Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology*, 2005.

[46] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Rob. Auton. Syst.*, 42(3-4), 2003.

[47] D. R. Forsyth. *Group dynamics*. T. Wadsworth, fourth edition, 2009.

[48] S. Fujii, K. Kudo, T. Ohtsuki, and S. Oda. Tapping performance and underlying wrist muscle activity of non-drummers, drummers, and the world's fastest drummer. *Neurosci. Lett.*, 459(2):69–73, 2009.

[49] D. Gatica-Perez. Automatic nonverbal analysis of social interaction in small groups: A review. *Image Vision Comput*, 2009.

[50] A. Gelblum, I. Pinkoviezky, E. Fonio, A. Ghosh, N. Gov, and O. Feinerman. Ant groups optimally amplify the effect of transiently informed individuals. *Nature communications*, 6, 2015.

[51] A. Gelman. Commentary: P values and statistical practice. *Epidemiology*, 24(1):69–72, 2013.

[52] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen. Development of a driver information and warning system with vehicle hardware-in-the-loop simulations. *Mechatronics*, 19(7):1091–1104, 2009.

[53] M. J. Gonzales and L. D. Riek. A sociable robotic aide for medication adherence. In *Proc. of International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, New York, NY, USA, 2012. ACM.

[54] E. C. Grigore, K. Eder, A. G. Pipe, C. Melhuish, and U. Leonards. Joint action understanding improves robot-to-human object handover. In *IEEE/RSJ Internation Conference on Intelligent Robots and Systems (IROS)*, 2013.

[55] G. Hager and H. Durrant-Whyte. Information and multi–sensor coordination. *Uncertainty in Artificial Intelligence*, 2, 1988.

[56] S. Hahn. Understanding noninferiority trials. *Korean J. Pediatr.*, 55(11):403–7, 2012.

[57] E. A. Hanushek, J. F. Kain, J. M. Markman, and S. G. Rivkin. Does peer ability affect student achievement? *J Appl Econom*, 2003.

[58] S. K. Hasnain, G. Mostafaoui, and P. Gaussier. A synchrony-based perspective for partner selection and attentional mechanism in human-robot interaction. *Paladyn*, 2013.

[59] C. J. Hayes, C. R. Crowell, and L. D. Riek. Automatic processing of irrelevant co-speech gestures with human but not robot actors. In *Proc. of ACM/IEEE HRI*, 2013.

[60] C. J. Hayes, M. Moosaei, and L. D. Riek. Exploring implicit human responses to robot mistakes in a learning from demonstration task. In *Robot and Human Interactive Communication (RO-MAN)*, 2016.

[61] H. Hennig. Synchronization in human musical rhythms and mutually interacting complex systems. *PNAS*, 2014.

[62] T. Himberg and M. R. Thompson. Learning and synchronising dance movements in south african songs: Cross-cultural motion-capture study. *Dance Res*, 2011.

[63] G. Hoffman and C. Breazeal. Collaboration in human-robot teams. In *AIAA Intelligent Systems Technical Conference*, 2007.

[64] G. Hoffman and C. Breazeal. Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics*, 2007.

[65] G. Hoffman and C. Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. *Human-Robot Interact.*, 2007.

[66] G. Hoffman and C. Breazeal. Anticipatory Perceptual Simulation for Human-Robot Joint Practice: Theory and Application Study. *AAAI*, 2008.

[67] G. Hoffman and G. Weinberg. Gesture-based human-robot jazz improvisation. In *IEEE Conf. on Robotics and Automation*, 2010.

[68] G. Hoffman and G. Weinberg. Synchronization in human-robot Musicianship. *19th Int. Symp. Robot Hum. Interact. Commun.*, pages 718–724, sep 2010.

[69] G. Hoffman and G. Weinberg. Synchronization in human-robot Musicianship. *RO-MAN*, 2010.

[70] G. Hoffman and G. Weinberg. Interactive improvisation with a robotic marimba player. *Auton. Robots*, 2011.

[71] G. Hoffman and G. Weinberg. Interactive improvisation with a robotic marimba player. *Auton. Robots*, 31(2-3):133–153, jun 2011.

[72] A. Hreljac and R. N. Marshall. Algorithms to determine event timing during normal walking using kinematic data. *J Biomech*, 2000.

[73] C.-m. Huang, M. Cakmak, and B. Mutlu. Adaptive Coordination Strategies for Human-Robot Handovers. In *Robot. Sci. Syst.*, 2015.

[74] T. Iqbal, M. J. Gonzales, and L. D. Riek. A Model for Time-Synchronized Sensing and Motion to Support Human-Robot Fluency. In *ACM/IEEE Human-Robot Interaction, Workshop on Timing in HRI*, 2014.

[75] T. Iqbal, M. J. Gonzales, and L. D. Riek. Mobile robots and marching humans: Measuring synchronous joint action while in motion. In *AAAI Fall Symp. on AI-HRI*, 2014.

[76] T. Iqbal, M. J. Gonzales, and L. D. Riek. Joint action perception to enable fluent human-robot teamwork. In *Proc. of IEEE Robot and Human Interactive Communication*, 2015.

[77] T. Iqbal, M. Moosaei, and L. D. Riek. Tempo Adaptation and Anticipation Methods for Human-Robot Teams. In *Robotics: Science and Systems, Planning for HRI: Shared Autonomy and Collab. Robotics Work.*, 2016.

[78] T. Iqbal, S. Rack, and L. D. Riek. Will you join the dance? toward synchronous joint action in human robot teams. In *Proc. of Joint Action Meeting*, 2015.

[79] T. Iqbal, S. Rack, and L. D. Riek. Movement coordination in human-robot teams: A dynamical systems approach. *IEEE Transactions on Robotics*, 32(4):909–919, 2016.

[80] T. Iqbal and L. Riek. Assessing group synchrony during a rhythmic social activity: A systemic approach. In *Proc. of the conference of the International Society for Gesture Studies (ISGS)*, 2014.

[81] T. Iqbal and L. D. Riek. Role distribution in synchronous human-robot joint action. In *Proc. of IEEE RO-MAN, Towards a Framework for Joint Action*, 2014.

[82] T. Iqbal and L. D. Riek. A Method for Automatic Detection of Psychomotor Entrainment. *IEEE Transactions on Affective Computing*, 2015.

[83] T. Iqbal and L. D. Riek. Detecting and Synthesizing Synchronous Joint Action in Human-Robot Teams. In *International Conference on Multimodal Interaction*, 2015.

[84] T. Iqbal and L. D. Riek. Human coordination dynamics with heterogeneous robots in a team. In *Proc. of ACM/IEEE Human-Robot Interaction Conference*, 2016.

[85] T. Iqbal and L. D. Riek. Leveraging coordination dynamics to enable human robot teams. In *Mid-West Robotics Workshop*, 2016.

[86] T. Iqbal and L. D. Riek. Can robots synchronize with humans in tempo changing environments? In *Proc. of Joint Action Meeting*, 2017.

[87] T. Iqbal and L. D. Riek. Coordination dynamics in multi-human multi-robot teams. *IEEE Robotics and Automation Letters (RA-L)*, 2017.

[88] T. Iqbal and L. D. Riek. Human robot teaming: Approaches from joint action and dynamical systems. *Humanoid Robotics: A Reference*, In review.

[89] N. Jarrassé, T. Charalambous, and E. Burdet. A framework to describe, analyze and generate interactive motor behaviors. *PLoS One*, 7(11):e49945, jan 2012.

[90] N. Jarrasse, V. Sanguineti, and E. Burdet. Slaves no longer: review on role assignment for human-robot joint motor action. *Adapt. Behav.*, 22(1):70–82, sep 2013.

[91] D. B. Jayagopi, H. Hung, C. Yeo, and D. Gatica-Perez. Modeling Dominance in Group Conversations Using Nonverbal Activity Cues. *IEEE T Audio Speech*, 2009.

[92] O. C. Jenkins, M. J. Matari, and S. Weber. Primitive-Based Movement Classification for Humanoid Imitation. *IEEE-RAS Int'l Conference on Humanoid Robotics*, 2000.

[93] S. Ji, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *Pami*, 35(1):221–31, 2013.

[94] T. Kanda, H. Ishiguro, M. Imai, and T. Ono. Body movement analysis of human-robot interaction. In *IJCAI*, 2003.

[95] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[96] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. *Comput. Vis. Pattern Recognit. (CVPR), 2014 IEEE Conf.*, pages 1725–1732, 2014.

[97] H. Kazerooni. Human-robot interaction via the transfer of power and information signals. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):450–463, 1990.

[98] P. E. Keller. Joint action in music performance. *Emerging Communication*, 10:205, 2008.

[99] O. Kermorgant and F. Chaumette. Multi-sensor data fusion in sensor-based control: Application to multi-camera visual servoing. *2011 IEEE Int. Conf. Robot. Autom.*, (3), May 2011.

[100] S. Khalfa, M. Roy, P. Rainville, S. Dalla Bella, and I. Peretz. Role of tempo entrainment in psychophysiological differentiation of happy and sad music? *J Psychophysiol*, 2008.

[101] A. K. Khalil, V. Minces, G. McLoughlin, and A. Chiba. Group rhythmic synchrony and attention in children. *Front. Psychol.*, 4(SEP):1–7, 2013.

[102] S. Kirschner and M. Tomasello. Joint drumming: social context facilitates synchronization in preschool children. *J Exp Child Psychol*, 2009.

[103] G. Klien, D. Woods, J. Bradshaw, R. Hoffman, and P. Feltovich. Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems*, 19(6), 2004.

[104] J. Koenemann, F. Burget, and M. Bennewitz. Real-time imitation of human whole-body motions by humanoids. In *IEEE Conf. on Robotics and Automation*, 2014.

[105] I. Konvalinka, P. Vuust, A. Roepstorff, and C. D. Frith. Follow you, follow me: continuous mutual prediction and adaptation in joint tapping. *J of Experimental Psychology*, 2010.

[106] I. Konvalinka, D. Xygalatas, J. Bulbulia, U. Schjødt, E.-M. Jegindø, S. Wallot, G. Van Orden, and A. Roepstorff. Synchronized arousal between performers and related spectators in a fire-walking ritual. *P Natl Acad Sci USA*, 2011.

[107] H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. *ICML*, 2013.

[108] H. S. Koppula, A. Jain, and A. Saxena. Anticipatory planning for human-robot teams. *Springer Tracts Adv. Robot.*, 2016.

[109] G. Kress. *Multimodality: A social semiotic approach to contemporary communication*. Routledge, 2009.

[110] T. Kreuz, J. S. Haas, A. Morelli, H. D. Abarbanel, and A. Politi. Measuring spike train synchrony. *J Neurosci Meth*, 2007.

[111] T. Kreuz, F. Mormann, R. G. Andrzejak, A. Kraskov, K. Lehnertz, and P. Grassberger. Measuring synchronization in coupled model systems: A comparison of different approaches. *Physica D: Nonlinear Phenomena*, 2007.

[112] G. Kwakkel, B. J. Kollen, and H. I. Krebs. Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review. *Neurorehabilitation and neural repair*, 2007.

[113] J. P. Lachaux, E. Rodriguez, J. Martinerie, and F. J. Varela. Measuring phase synchrony in brain signals. *Hum Brain Mapp*, 1999.

[114] D. Lakens. Movement synchrony and perceived entitativity. *J Exp Soc Psychol*, 2010.

[115] D. Lakens and M. Stel. If They Move in Sync, They Must Feel in Sync: Movement Synchrony Leads to Attributions of Rapport and Entitativity. *Soc Cognition*, 2011.

[116] S. Lallée, U. Pattacini, J. D. Boucher, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, K. Hamann, and J. Steinwender. Towards a platform-independent cooperative human-robot interaction system: Ii. perception, execution and imitation of goal directed actions. In *International Conference on Intelligent Robots and Systems*, 2011.

[117] E. W. Large. Resonating to musical rhythm: theory and experiment. *Psychology of time*, pages 189–232, 2008.

[118] M. Lavelle, P. G. Healey, and R. McCabe. Is nonverbal communication disrupted in interactions involving patients with schizophrenia? *Schizophrenia bulletin*, 2013.

[119] M. Lavelle, P. G. Healey, and R. McCabe. Participation during first social encounters in schizophrenia. *PloS one*, 2014.

[120] I. Leite, G. Castellano, A. Pereira, C. Martinho, and A. Paiva. Modelling empathic behaviour in a robotic game companion for children: an ethnographic study in real-world settings. *ACM/IEEE Int. Conf. Human-Robot Interact.*, 2012.

[121] N. E. Leonard, G. Young, K. Hochgraf, D. Swain, A. Trippe, W. Chen, and S. Marshall. In the dance studio: Analysis of human flocking. In *American Control Conference (ACC), 2012*, 2012.

[122] N. E. Leonard, G. F. Young, K. Hochgraf, D. T. Swain, A. Trippe, W. Chen, K. Fitch, and S. Marshall. In the Dance Studio: An Art and Engineering Exploration of Human Flocking. *Control. Art*, 2014.

[123] H. C. Lin, I. Shafran, D. Yuh, and G. D. Hager. Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions. *Comput. Aided Surg.*, 11(5):220–30, sep 2006.

[124] T. Lourens, R. van Berkel, and E. Barakova. Communicating emotions and mental states to robots in a real time parallel framework using Laban movement analysis. *Rob. Auton. Syst.*, 58(12), 2010.

[125] M. M. Louwerse, R. Dale, E. G. Bard, and P. Jeuniaux. Behavior matching in multimodal communication is synchronized. *Cognitive Sci*, 2012.

[126] J. H. Macdonald. Lateral excitation of bridges by balancing pedestrians. *Proc. of the Royal Society A: Math., Phys. and Engg. Sci.*, 2009.

[127] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013.

[128] R. A. Mann and J. Hagy. Biomechanics of walking, running, and sprinting. *The American journal of sports medicine*, 1980.

[129] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 2007.

[130] N. Marwan and J. Kurths. Nonlinear analysis of bivariate data with cross recurrence plots. *Phys Lett A*, 2002.

[131] J. Mates. A model of synchronization of motor acts to a stimulus sequence. *Biological cybernetics*, 70(5):463–473, 1994.

[132] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2009.

[133] M. P. Michalowski, S. Sabanovic, and H. Kozima. A dancing robot for rhythmic social interaction. In *Proc. of ACM/IEEE HRI*, 2007.

[134] D. Mills. Internet time synchronization: the network time protocol. *IEEE Trans. Commun.*, 39(10), 1991.

[135] A. Moon, D. M. Troniak, B. Gleeson, M. K. X. J. Pan, M. Zheng, B. A. Blumer, K. MacLean, and E. A. Croft. Meet Me Where I'M Gazing: How Shared Attention Gaze Affects Human-robot Handover Timing. In *ACM/IEEE Int. Conf. Human-robot Interact.*, 2014.

[136] M. Moosaei, C. J. Hayes, and L. D. Riek. Performing facial expression synthesis on robot faces: A real-time software system. In *Proceedings of the 4th International AISB Symposium on New Frontiers in Human-Robot Interaction*, 2015.

[137] L. Y. Morales Saiki, S. Satake, R. Huq, D. Glas, T. Kanda, and N. Hagita. How do people walk side-by-side?: using a computational model of human behavior for a social robot. In *Proc. of ACM/IEEE HRI*, 2012.

[138] A. Mörtl, T. Lorenz, and S. Hirche. Rhythm patterns interaction-synchronization behavior for human-robot joint action. *PloS one*, 2014.

[139] B. Mutlu, A. Terrell, and C.-M. Huang. Coordination mechanisms in human-robot collaboration. In *Proc. of ACM/IEEE HRI*, 2013.

[140] M. Nagy, Z. Ákos, D. Biro, and T. Vicsek. Hierarchical group dynamics in pigeon flocks. *Nature*, 2010.

[141] M. Nagy, G. Vásárhelyi, B. Pettit, I. Roberts-Mariani, T. Vicsek, and D. Biro. Context-dependent hierarchies in pigeons. *P Natl Acad Sci*, 2013.

[142] National Robotics Week. *http://engineering.nd.edu/NDNRW*, Dec. 2013.

[143] Z. Néda, E. Ravasz, Y. Brechet, T. Vicsek, and A.-L. Barabási. Self-organizing processes: The sound of many hands clapping. *Nature*, 2000.

[144] G. Neumann, C. Daniel, and A. Kupcsik. Information-Theoretic Motor Skill Learning. *AAAI 2013 Workshop on Intelligent Robotic System*, 2013.

[145] B. Ni, G. Wang, and P. Moulin. RGBD-HuDaAct: A Color-Depth Video Database for Human Daily Activity Recognition. *Consum. Depth Cameras Comput. Vis.*, 2013.

[146] S. Niekum and S. Chitta. Incremental Semantically Grounded Learning from Demonstration. *Robot. Sci. Syst. IX*, 2013.

[147] A. Nigam and L. Riek. Social context perception for mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 1–6, 2015.

[148] S. Nikolaidis, K. Gu, R. Ramakrishnan, J. Shah, and R. O. May. Efficient Model Learning for Human-Robot Collaborative Tasks. pages 1–9, 2014.

[149] S. Nikolaidis, P. Lasota, G. Rossano, C. Martinez, T. Fuhlbrigge, and J. Shah. Human-robot collaboration in manufacturing: Quantitative evaluation of predictable, convergent joint action. In *ISR*, 2013.

[150] L. Noy, E. Dekel, and U. Alon. The mirror game as a paradigm for studying the dynamics of two people improvising motion together. *P Natl Acad Sci USA*, 2011.

[151] M. O'Connor and L. Riek. Detecting social context: A method for social event classification using naturalistic multimodal data. *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.

[152] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley MHAD: A comprehensive Multimodal Human Action Database. *2013 IEEE Work. Appl. Comput. Vis.*, 2013.

[153] K. Ong, G. Seet, and S. Sim. An Implementation of Seamless Human-Robot Interaction for Telerobotics. *Int. J. Adv. Robot. Syst.*, 5(2):167–176, 2008.

[154] K. W. Ong, G. Seet, and S. K. Sim. Sharing and trading in a human-robot system. *Cutting Edge Robotics, V. Kordic, A. Laznica and M. Merdan*, 2005.

[155] H. W. Park and A. M. Howard. Understanding a child's play for robot interaction by sequencing play primitives using hidden markov models. In *Proc. IEEE Int. Conf. Robot. Autom.*, 2010.

[156] E. Pereda, R. Q. Quiroga, and J. Bhattacharya. Nonlinear multivariate analysis of neurophysiological signals. *Prog Neurobiol*, 2005.

[157] C. Pérez-DArpino and J. Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *IEEE Conf. on Robotics and Automation*, 2015.

[158] M. A. Pérez-Quiñones and J. L. Sibert. A collaborative model of feedback in human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1996.

[159] J. Phillips-Silver, C. Aktipis, and G. Bryant. The ecology of entrainment: Foundations of coordinated rhythmic movement. *Music Percept*, 2010.

[160] G. Piaggio, D. R. Elbourne, D. G. Altman, S. J. Pocock, S. J. W. Evans, and for the CONSORT Group. Reporting of Noninferiority and Equivalence Randomized Trials. *Jama*, 295(10):1152, 2006.

[161] R. Poppe. A survey on vision-based human action recognition. *Image Vis. Comput.*, 2010.

[162] K. Prepin and P. Gaussier. How an agent can detect and use synchrony parameter of its own interaction with a human? In *Dev. of Multimodal Interfaces: Active Listening and Synchrony*. 2010.

[163] C. G. Prince, G. J. Hollich, N. A. Helder, E. J. Mislivec, A. Reddy, S. Salunke, and N. Memon. Taking synchrony seriously: a perceptual-level model of infant synchrony detection. In *Int. Workshop Epigenetic Robot.*, 2004.

[164] R. Q. Quiroga, A. Kraskov, T. Kreuz, and P. Grassberger. Performance of different synchronization measures in real data: a case study on electroencephalographic signals. *Phys Rev E*, 2002.

[165] R. Q. Quiroga, T. Kreuz, and P. Grassberger. Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Phys Rev E*, 2002.

[166] S. Rack, T. Iqbal, and L. Riek. Enabling synchronous joint action in human-robot teams. In *Proc. of ACM/IEEE Human-Robot Interaction*, 2015.

[167] B. Raducanu, J. Vitria, and D. Gatica-Perez. You are fired! nonverbal role analysis in competitive meetings. In *Proc. of IEEE ICASSP*, 2009.

[168] M. Ramanathan, W.-y. Yau, and E. K. Teoh. Human Action Recognition With Video Data : Research and Evaluation Challenges. *IEEE Transactions on Human-Machine Systems*, 2014.

[169] B. H. Repp. Automaticity and voluntary control of phase correction following event onset shifts in sensorimotor synchronization. *Journal of Experimental Psychology: Human Perception and Performance*, 28(2):410, 2002.

[170] B. H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic bulletin & review*, 12(6):969–992, 2005.

[171] B. H. Repp and P. E. Keller. Sensorimotor synchronization with adaptively timed sequences. *Human movement science*, 27(3):423–456, 2008.

[172] M. J. Richardson, R. L. Garcia, T. D. Frank, M. Gergor, and K. L. Marsh. Measuring group synchrony: a cluster-phase method for analyzing multivariate movement time-series. *Frontiers in Physiology*, 2012.

[173] L. Riek, T.-C. Rabinowitch, P. Bremner, A. Pipe, M. Fraser, and P. Robinson. Cooperative gestures: Effective signaling for humanoid robots. *ACM/IEEE Int. Conf. Human-Robot Interact.*, 2010.

[174] L. D. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1), 2012.

[175] L. D. Riek. Embodied computation: An active-learning approach to mobile robotics education. *IEEE T Educ*, 2013.

[176] L. D. Riek. The social co-robotics problem space: Six key challenges. In *In Proc. of Robotics: Science and Systems, Robotics Challenges and Visions Workshop*, 2013.

[177] L. D. Riek, M. F. Oconnor, and P. Robinson. Guess what? a game for affective annotation of video using crowd sourcing. In *Proc. of Affective Computing and Intelligent Interaction*. 2011.

[178] L. D. Riek, P. C. Paul, and P. Robinson. When my robot smiles at me: Enabling human-robot rapport via real-time head gesture mimicry. *J. Multimodal User Interfaces*, 2010.

[179] L. D. Riek, P. C. Paul, and P. Robinson. When my robot smiles at me: Enabling human-robot rapport via real-time head gesture mimicry. *Journal on Multimodal User Interfaces*, 3(1-2), 2010.

[180] L. D. Riek, T. C. Rabinowitch, P. Bremner, A. G. Pipe, M. Fraser, and P. Robinson. Cooperative gestures: Effective signaling for humanoid robots. In *Proc. of ACM/IEEE HRI*, 2010.

[181] L. D. Riek and P. Robinson. Real-time empathy: Facial mimicry on a robot. In *International Conference on Multimodal Interfaces., Affective Interaction in Natural Environments (AFFINE)*, 2008.

[182] L. D. Riek and P. Robinson. Challenges and opportunities in building socially intelligent machines [social sciences]. *IEEE Signal Processing Magazine*, 2011.

[183] R. Rienks and D. Heylen. Dominance detection in meetings using easily obtainable features. In *Machine Learning for Multimodal Interaction*. 2006.

[184] Robot Operating System. *http://www.ros.org*, May 2015.

[185] M. Rolf, M. Hanheide, and K. J. Rohlfing. Attention via synchrony: Making use of multimodal cues in social learning. *IEEE T Autonomous Mental Development*, 2009.

[186] M. C. Romano, M. Thiel, J. Kurths, I. Z. Kiss, and J. L. Hudson. Detection of synchronization for non-phase-coherent and non-stationary data. *Europhys Lett*, 2005.

[187] R. Ros, I. Baroni, and Y. Demiris. Adaptive human-robot interaction in sensorimotor task instruction: From human to robot dance tutors. *Rob. Auton. Syst.*, 2014.

[188] ROS Topic. *http://wiki.ros.org/rostopic*, June 2017.

[189] J. A. Ross. The reliability, validity, and utility of self-assessment. *Practical Assessment Research & Evaluation*, 2006.

[190] M. D. Rothmann, B. L. Wiens, and I. S. F. Chan. *Design and Analysis of Non-Inferiority Trials*. CRC Press, fourth edition, 2012.

[191] V. Roto and A. Oulasvirta. Need for non-visual feedback with long response times in mobile hci. In *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 2005.

[192] M. Ryoo, T. J. Fuchs, L. Xia, J. Aggarwal, and L. Matthies. Robot-centric activity prediction from first-person videos: What will they do to me? In *Proc. of ACM/IEEE HRI*, 2015.

[193] M. S. Ryoo and L. Matthies. First-Person Activity Recognition: What Are They Doing to Me? *In Proc. of IEEE Computer Vision and Pattern Recognition*, 2013.

[194] G. Salvendy. Classification of human motions. *Theoretical Issues in Ergonomics Science*, 2004.

[195] A. A. Samadani, A. Ghodsi, and D. Kulić. Discriminative functional analysis of human movements. *Pattern Recognit. Lett.*, 2013.

[196] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 2003.

[197] R. A. Schmidt. A schema theory of discrete motor skill learning. *Psychological review*, 1975.

[198] R. C. Schmidt and M. J. Richardson. Dynamics of interpersonal coordination. *Und Com Sys*, 2008.

[199] R. I. Schubotz. Prediction of external events with our motor system: towards a new framework. *Trends in cognitive sciences*, 11(5):211–218, 2007.

[200] J. Schumi and J. T. Wittes. Through the looking glass: understanding non-inferiority. *Trials*, 12(1):106, 2011.

[201] N. Sebanz, H. Bekkering, and G. Knoblich. Joint action: bodies and minds moving together. *T. Cogn. Sci.*, 2006.

[202] A. Semjen, D. Vorberg, and H.-H. Schulze. Getting synchronized with the metronome: Comparisons between phase and period correction. *Psychological Research*, 61(1):44–55, 1998.

[203] J. Shah and C. Breazeal. An Empirical Analysis of Team Coordination Behaviors and Action Planning With Application to Human-Robot Teaming. *Hum. Factors J. Hum. Factors Ergon. Soc.*, 2010.

[204] J. Shah, J. Wiken, B. Williams, and C. Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. *Proc. 6th Int. Conf. Human-robot Interact.*, 2011.

[205] C. Shi, M. Shiomi, C. Smith, T. Kanda, and H. Ishiguro. A Model of Distributional Handing Interaction for a Mobile Robot. *Robot. Sci. Syst.*, 2013.

[206] E. A. Sisbot and R. Alami. A human-aware manipulation planner. *IEEE Trans. Robot.*, 2012.

[207] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinavasa, M. Cakmak, and V. Micelli. Towards Seamless Human-Robot Handovers. *J. Human-Robot Interact.*, 2012.

[208] J. Stuckler and S. Behnke. Following human guidance to cooperatively carry a large object. *2011 11th IEEE-RAS Int. Conf. Humanoid Robot.*, pages 218–223, oct 2011.

[209] J. Sung, H. I. Christensen, and R. E. Grinter. Robots in the wild: Understanding long-term use. *Human-Robot Interact. (HRI), 2009 4th ACM/IEEE Int. Conf.*, pages 45–52, 2009.

[210] D. Tetteroo, A. Shirzad, M. S. Pereira, M. Zwinderman, D. Le, and E. Barakova. Mimicking expressiveness of movements by autistic children in game play. In *2012 International Confernece on Social Computing (SocialCom)*. IEEE, 2012.

[211] A. Thomaz, G. Hoffman, and M. Cakmak. Computational Human-Robot Interaction. *Foundations and Trends in Robotics*, 2016.

[212] A. L. Thomaz and C. Chao. Turn-taking based on information flow for fluent human-robot interaction. *AI Magazine*, 32(4), 2011.

[213] Turtlebot Robot. *www.willowgarage.com/turtlebot*, Feb. 2014.

[214] V. V. Unhelkar, C. Pérez-DArpino, L. Stirling, and J. Shah. Human-robot co-navigation using anticipatory indicators of human walking motion. In *IEEE Conf. on Robotics and Automation*, 2015.

[215] M. C. M. van der Steen and P. E. Keller. The ADaptation and Anticipation Model (ADAM) of sensorimotor synchronization. *Front. Hum. Neurosci.*, 2013.

[216] M. M. van der Steen, N. Jacoby, M. T. Fairhurst, and P. E. Keller. Sensorimotor synchronization with tempo-changing auditory sequences: Modeling temporal adaptation and anticipation. *Brain research*, 2015.

[217] N. R. van Ulzen, C. J. Lamoth, A. Daffertshofer, G. R. Semin, and P. J. Beek. Characteristics of instructed and uninstructed interpersonal coordination while walking side-by-side. *Neuroscience letters*, 432(2):88–93, 2008.

[218] G. Varni, G. Volpe, and A. Camurri. A System for Real-Time Multimodal Analysis of Nonverbal Affective Social Interaction in User-Centric Media. *IEEE T Multimedia*, 2010.

[219] G. Varni, G. Volpe, and B. Mazzarino. Towards a social retrieval of music content. In *Proc. IEEE Conf. on Social Computing, Privacy, Security, Risk and Trust*, 2011.

[220] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 2009.

[221] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D'Errico, and M. Schroeder. Bridging the gap between social animal and unsocial machine: A survey of social signal processing. *IEEE T on Affective Computing*, 2012.

[222] C. L. Webber Jr and J. P. Zbilut. Recurrence quantification analysis of nonlinear dynamical systems. *Tutorials in contemporary nonlinear methods for the behavioral sciences*, 2005.

[223] R. Wilcox, S. Nikolaidis, and J. A. Shah. Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing. In *Robotics: Science and Systems*, 2012.

[224] S. T. Witt, A. R. Laird, and M. E. Meyerand. Functional neuroimaging correlates of finger-tapping task variations: an ale meta-analysis. *Neuroimage*, 42(1):343–356, 2008.

[225] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Inf. Fusion*, 3(2), 2002.

[226] D. Xygalatas, I. Konvalinka, J. Bulbulia, and A. Roepstorff. Quantifying collective effervescence: Heart-rate dynamics at a fire-walking ritual. *Commun Integr Biol*, 2011.

[227] K. Yun and J. Honorio. Two-person interaction detection using body-pose features and multiple instance learning. *IEEE Comput. Vis. Pattern Recognit. Work. (CVPRW)*, 2012.

[228] J. P. Zbilut and C. L. Webber. Embeddings and delays as derived from quantification of recurrence plots. *Phys Lett A*, 1992.

[229] H. Zhang and L. E. Parker. 4-Dimensional Local Spatio-Temporal Features for Human Activity Recognition. *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2011.