2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Distributed Model Predictive Formation Control with Gait Synchronization for Multiple Quadruped Robots

Shaohang Xu[1,2], Wentao Zhang[1], Lijun Zhu[1] and Chin Pang Ho[2]

Abstract— **In this paper, we present a fully distributed framework for multiple quadruped robots in environments with obstacles. Our approach utilizes Model Predictive Control (MPC) and multi-robot consensus protocol to obtain the distributed control law. It ensures that all the robots are able to avoid obstacles, navigate to the desired positions, and meanwhile synchronize the gaits. In particular, via MPC and consensus, the robots compute the optimal trajectory and the contact profile of the legs. Then an MPC-based locomotion controller is implemented to achieve the gait, stabilize the locomotion and track the desired trajectory. We present experiments in simulation and with three real quadruped robots in an environment with a static obstacle.**

I. Introduction

The advanced mobility of quadruped robots motivates a lot of works on the development of quadruped machines, such as ANYmal [1], HyQ [2], MIT Cheetah series [3]–[6] from the robotics research community, and Boston Dynamics Spot [7] and Unitree A1 [8] from the commercial robot companies. Recently, there has been an increasing number of research works on locomotion control for an individual robot, towards highly dynamic and stable quadruped locomotion in difficult and unstructured terrains [9]–[16]. However, the isolated operation of individual robots limits the autonomy, perception, and decision-making [17]. Instead, the cooperation of multiple robots can improve the applicability in complex tasks such as search and rescue, industrial inspection, etc [18]–[21]. For example, multiple quadruped robots can explore subterranean environments autonomously [22], [23], which are inhospitable to humans.

This paper considers the locomotion control of multiple quadruped robots in a distributed way. In particular, we propose a distributed control framework to achieve the following objectives:

- All the robots are regulated to their own desired positions with a certain formation;
- All the robots are able to avoid collisions within the robot swarm or against the obstacles in the environment;
- The gaits of all the robots are synchronized;
- All the robots stabilize their dynamic locomotion and do not lose balance.

An illustrating example is shown in Fig. 1. Next, we give a brief review on the related work and highlight our main contribution.

[1] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China, shaohangxu@hust.edu.cn, wentaozhang@hust.edu.cn, ljzhu@hust.edu.cn
[2] School of Data Science, City University of Hong Kong, HKSAR, shaohanxu2-c@my.cityu.edu.hk, clint.ho@cityu.edu.hk

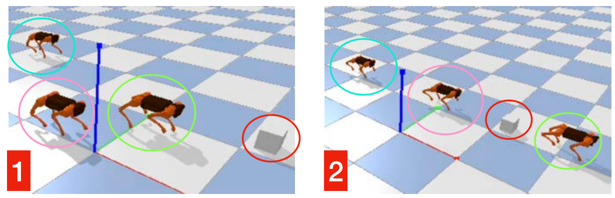


Fig. 1. An illustrating example of our control objective. The left and right figures show the test scenarios in the simulator PyBullet [24] before and after distributed control, respectively. The red circle highlights the obstacle in the environment. Robot 1, 2, and 3 are circled in the green, pink, and blue colors, respectively. Our first two objectives are controlling all the robots to achieve a predefined formation (a queue in this example) and meanwhile avoid any collision within the robot swarm or against the obstacle. Moreover, a significant objective in this paper is to synchronize the gaits of all the robots, which can be understood straightforward by observing the leg states of the robots: the robots have different gaits in the left figure while their leg states become the same after distributed control. Last but not the least, the dynamic locomotion should be stabilized, i.e., all the robots should not lose any balance to achieve the above three objectives.

A. Related Work

1) Swarm Robotics: Inspired by the collective behaviors of social animals such as ants [25], bees [26], birds and fish [27], researchers are interested in developing aerial, underwater, and ground swarm robotic systems to achieve the similar capability of their animal counterparts [28]–[31]. Compared with wheeled or tracked vehicles, legged robots are able to traverse difficult and unstructured terrains with dynamic and agile locomotion [32]. However, there are few works focused on legged robots in the swarm robotics community. Indeed, the swarm of quadruped robots shows extraordinary performance in real-world applications, such as exploration of underground environments [22], [23]. Nevertheless, the unique locomotion modal of legged robots, i.e., gait, is rarely considered for the swarm robotic system. By synchronizing the gait, the quadruped robot swarm is expected to accomplish some particular tasks better, such as coordinated carrying loads [33]. In this paper, we propose a dynamics model for quadrupedal gaits and then present a consensus-based control method for gait synchronization. The main advantage of the proposed model is that, it can generate the periodic gaits which are easy to stabilize on real robots. Moreover, the distributed control system with the proposed controller is guaranteed to converge.

2) Distributed Formation Control: There is a large number of works on formation control of multiple robots in environments with obstacles [34]–[37]. In this problem, a significant challenge is that there are always physical or task constraints, such as the limit of the control input,

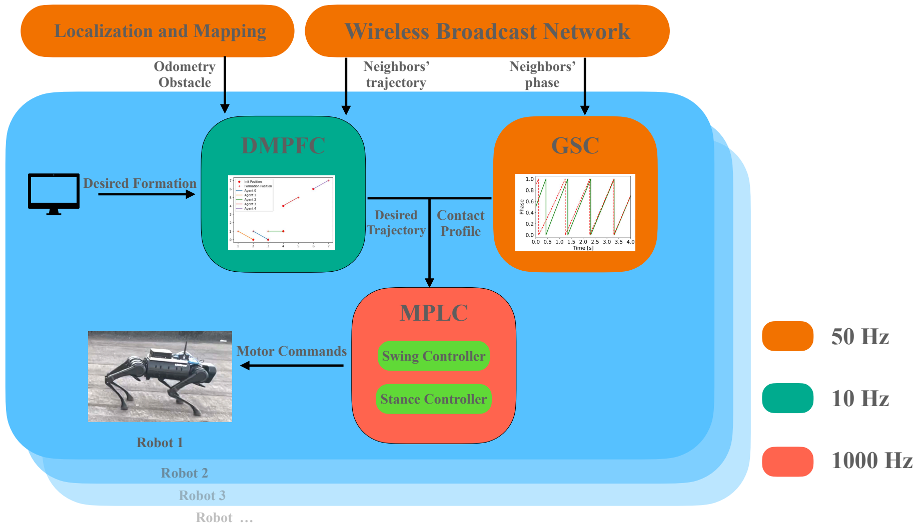


Fig. 2. An overview of our control framework. We assume the robot is equipped with localization and mapping modules, which can provide DMPFC with the odometry and the obstacle information. Moreover, wireless broadcast network is also required by our controller, through which the robots can exchange their trajectory and gait phase information. DMPFC outputs the desired trajectory with a predefined formation, and GSC outputs the desired contact profile. Finally, MPLC calculates the desired motor torque commands to the low level motor controllers on the robot, which is to stabilize the dynamic locomotion and achieve the desired gait. The color indicates the update frequency of the corresponding modules on our hardware.

collision avoidance, etc. In control theory, there are many strategies to handle constraints, but in particular, MPC is one of the most attractive methods because it is able to straightforwardly address constraints and meanwhile has the optimal performance [38], [39]. For distributed control of multi-agent systems, distributed MPC theory is well-studied in literature [40]–[42]. Compared with centralized MPC, distributed MPC is expected to be more scalable in structures and with lower communication costs [43]. By decomposing formation control from locomotion control, we implement distributed MPC in our proposed framework to obtain the optimal trajectory for each robot. By simplifying the computation and omitting the stability analysis, we use off-the-shelf solvers [44], [45] to solve a nonlinear, nonconvex program online. The proposed method can be easily combined with other modules in the framework and deployed on real robots.

3) Quadruped Locomotion Control: Locomotion control of an individual robot has been learned in the quadruped locomotion community, and it can be categorized into learning-based methods [12]–[16] or model-based methods [9]–[11]. Learning-based methods do not require the dynamics model of quadruped robots and show extraordinary adaptiveness, robustness, and stability in complex environments [12]–[14]. Although some recent approaches can deal with multi-modal gaits [16], [46], [47], most learning-based methods cannot adjust the contact schedule online due to the lack of model representation. Therefore, the gaits of multiple quadruped robots cannot be synchronized in this way. On the contrary, model-based methods can handle multi-modal gaits straightforwardly, as long as the contact profile is predefined. In particular, MPC-based methods are proposed to stabilize highly dynamic locomotion [10], [48], and all common gaits can be achieved on real robots [9].

B. Contribution

In this paper, we propose a distributed control framework for multiple quadruped robots to achieve formation control with gait synchronization, visualized in Fig. 2. The proposed method decomposes the overall objective and combines three sub-controllers: Distributed Model Predictive Formation Controller (DMPFC), Gait Synchronization Controller (GSC), and Model Predictive Locomotion Controller (MPLC). The main contribution is that our method extends the capabilities of legged robots to coordinate in a distributed way. To the best of our knowledge, this is the first time that a distributed control framework is applied to a quadruped robot swarm with gait synchronization. The proposed control framework is validated on multiple quadruped robots in both simulation and the real world.

II. Methodology

In this section, the distributed control framework is presented in detail. First, DMPFC is introduced to design the optimal trajectory. Then, the dynamics model of quadrupedal gaits is first proposed and GSC is then designed based on the consensus protocol. Finally, MPLC is briefly introduced for the sake of completeness, in order to track the desired trajectory and to stabilize the generated gait.

A. Distributed Model Predictive Formation Control

The control objective of DMPFC is to regulate all the robots to desired positions with a certain formation. Meanwhile, all the robots are able to avoid collisions and obstacles. In order to make the problem tractable, we omit the whole-body dynamics but use a simple difference model,

$$s_i(k+1) = s_i(k) + v_i(k), \tag{1}$$

where $s_i \triangleq [x_i, y_i, \gamma_i]^\top$ is the state vector of the i-th robot with x_i, y_i, γ_i representing the positions and yaw angle in the global two-dimensional frame; $v_i \triangleq [v_{x,i}, v_{y,i}, v_{\gamma,i}]$ is the control input with $v_{x,i}, v_{y,i}, v_{\gamma,i}$ representing the desired velocity of x_i, y_i, γ_i to be tracked in the following MPLC module.

For the i-th robot at time step k, the following optimization problem is solved in DMPFC:

$$\begin{aligned}\min_{v_i(\cdot|k)} \sum_{l=0}^{N-1} (\rho_i \|v_i(k+l|k)\|_2^2 \\ + \alpha_i \sum_{j\in\mathcal{N}_i} \|s_j(k+l|k) - s_i(k+l|k) - d_{ij}\|_2^2) \\ + \beta_i \|s_i^d - s_i(k+l|k)\|_2^2,\end{aligned} \tag{2}$$

$$\text{s.t.} s_i(k|k) = s_i(k) \tag{3}$$

$$s_i(k+1+l|k) = s_i(k+l|k) + v_i(k+l|k), \tag{4}$$

$$\|s_j(k+l+1|k) - s_i(k+l+1|k)\|_2^2 \geq R, \tag{5}$$

$$\|s_i(k+l+1|k) - s_o(k+l+1|k)\|_2^2 \geq D, \tag{6}$$

$$v_i(k) \in \mathbb{U} \tag{7}$$

$$l = 0, 1, \cdots, N-1, \ j \in \mathcal{N}_i, \ o \in \mathcal{O},$$

where the notation $a(k+l|k)$ represents the value of a at the time step $k+l$ which is predicted at the time step k, $\rho_i, \alpha_i, \beta_i$ are positive weights; $\mathcal{N}_i$ denotes the set of the neighbors on the communication topology that is an undirected graph in this paper; s_i^d is the desired position of robot i; $d_{ij} \triangleq s_j^d - s_i^d$ is the desired relative position between robot i and j; $s_o \triangleq [x_o, y_o]^\top$ is the positions of the o-th obstable with o in the set $\mathcal{O}$; R and D are the predefined safety margin for group self-collision and obstacle avoidance.

The motivation of the optimization problem is illustrated as follows. The optimization objective (2) is to penalize the energy cost and the differences between the predictive and desired states in the formation. The equality constraint (3) specifies the initial states and (4) represents the dynamics model (1). The inequality constraint (5) is to avoid collisions among the quadruped robots, while (6) is introduced for obstacle avoidance. The set $\mathbb{U}$ in (7) gives the velocity limitation and defined as $\{v \in \mathbb{R}^3 | \underline{v} \leq v \leq \overline{v}\}$ where $\underline{v}$ and $\overline{v}$ are the lower and upper bound respectively.

In this paper, we assume that s_o, s_i^d, d_{ij}, and $\mathcal{N}_i$ are known for robot i. In practice, all the above-mentioned information can be obtained through the localization and mapping module or wireless communication. In this paper, we aim to apply the proposed controller onto the quadruped robot swarm, while the design of the terminal set and the terminal controller is omitted for brevity, and the analysis of recursive feasibility and stability will be left for the future work.

The above optimization problem is nonlinear and non-convex, and it is challenging to obtain the globally optimal solution. Instead, we use an interior point solver [44] to quickly obtain the local optimum. It is known that a good initial guess could lead to high-quality local optimum and improve the solving frequency. In this paper, we delete the constraints (5) and (6) and solve the sub-problem. Compared to the original problem, the sub-problem is a convex quadratic program and can be solved efficiently by many solvers. We use [45] to obtain the optimal solution of the sub-problem and then use it to initialize the original nonconvex program. In practice, the overall solving frequency is fast enough for the formation control, and the control performance is satisfactory in our experiments.

B. Gait Synchronization Control

Here we introduce the parameterization method for periodic gaits of quadruped robots. The key parameters are as follows:

- Stepping frequency f. The stepping frequency is used to represent how fast a gait cycle is. It is equivalent to the inverse of the periodic time of the gait. It is known that when f is high, the locomotion would be more robust. However, a higher f might be less energy efficient for low-speed locomotion. This is caused by frequent but not necessary leg swings.
- Nominal phase $\phi(t)$, phase offset ϕ_{offset} and phase threshold $\phi_{\text{threshold}}$. The nominal phase ϕ is defined as

$$\phi(t) = (f_0 t) \bmod (1), \tag{8}$$

where f_0 is the nominal stepping frequency. All legs of an individual robot share the same nominal phase, but they have different offsets in order to capture different kinds of gaits:

$$\phi^l = \phi + \phi^l_{\text{offset}}, \tag{9}$$

where ϕ^l denotes the phase of the l-th leg and ϕ^l_{offset} denotes its corresponding offset to the nominal phase. When $\phi^l_{\text{offset}} \geq \phi_{\text{threshold}}$, the l-th leg is in the swinging phase according to the contact schedule. Otherwise, it is in the stance phase.

In this paper, we only consider the case that the l-th legs of all the robots share the same ϕ^l_{offset} and $\phi_{\text{threshold}}$. Then, when the nominal phase ϕ is synchronized, the movement of all legs are synchronized. Moreover, we appoint the same nominal stepping frequency f_0 for all the robots.

To synchronize the gaits, we define our control input as an offset f_i to the nominal stepping frequency. Then, we can obtain the model of the gait for robot i as follows,

$$\phi_i = (\phi_i(0) + (f_0 + f_i)t) \bmod (1), \tag{10}$$

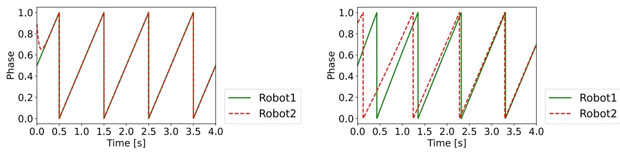
where ϕ_i is the nominal phase and $\phi_i(0)$ is its initial value; f_i is the offset to the nominal stepping frequency. f_i is used to regulate the internal clock, which is defined as

$$\theta_i = \phi_i(0) + (f_0 + f_i)t, \tag{11}$$

The robots transmit their internal clocks to their neighbors through wireless communication. Then, the distributed feedback controller of gait synchronization is to design f_i for the synchronization of θ_i.

In this paper, we use the consensus protocol [49] without time delays, meaning that the control law is designed as

$$f_i = -\sum_{j\in\mathcal{N}_i} a_{ij}(\theta_i - \theta_j) \tag{12}$$



(a) (b)

Fig. 3. An illustrating example of different dynamics models for gait synchronization. (a) The dynamics model (15) with the controller (16). (b) Our method. There are two robots in this example: the initial phase of Robot 2 is 0.9 while the initial phase of Robot 1 is 0.5. The consensus protocol is used for both models. We can observe that the robots in (a) achieve consensus faster than that in (b). However, in (a), Robot 2 decreases its phase from the initial value 0.9 to nearly 0.6, which will lead to undesired, challenging gaits to stabilize. In (b), the phase trajectory using our method remains periodic and smooth, and the generated gait is more natural and easier to stabilize using MPLC.

where $a_{ij} = a_{ji}$ are positive weights for $i \neq j$. The convergence is given as follows.

Theorem 1: Assume the communication topology is a connected graph. Then, when the time is sufficiently large, the gaits of the quadruped robot swarm are synchronized, i.e., when $t \to \infty$, $\phi_1 = \phi_2 = \cdots = \phi_n$.

Proof: We substitute (12) into (11), and then we can obtain the dynamics model for the gaits of the robots,

$$\dot{\theta}_i = f_0 - \sum_{j \in \mathcal{N}_i} a_{ij}(\theta_i - \theta_j). \tag{13}$$

Define $\bar{\theta}_i \triangleq \theta_i - f_0 t$. Then we have

$$\dot{\bar{\theta}}_i = - \sum_{j \in \mathcal{N}_i} a_{ij}(\bar{\theta}_i - \bar{\theta}_j) \tag{14}$$

From Theorem 5 and 6 in [49], we can prove that when the time is sufficiently large, $\dot{\bar{\theta}}_i$ will converge to zero for all i, and $\bar{\theta}_i$ will converge to be identical for all i. Then, from the definition of $\bar{\theta}_i$, θ_i will be identical for all i, i.e., all the robots will get the internal clock consensus. Therefore their nominal phases ϕ_i also achieve consensus due to (10) and their gaits are synchronized. ■

Intuitively, the robots achieve the phase consensus by accelerating or decelerating their stepping frequencies. The main advantage of the proposed model and controller for gait synchronization is that the generated gait is easier to stabilize. For example, another way is to use an offset to the internal clock as the control input, denoted as δ_i. Then the dynamics model is

$$\theta_i = \phi_i(0) + f_0 t + \delta_i \tag{15}$$

We can also utilize the consensus protocol to synchronize the gaits:

$$\delta_i = - \sum_{j \in \mathcal{N}_i} a_{ij}(\theta_i - \theta_j) \tag{16}$$

Similar to Theorem 1, we can also prove the convergence of the control system. However, the synchronization procedure will lead to undesired phase trajectory, which is challenging for locomotion controllers. Please see the illustrating example shown as Fig. 3 for more details.

It is worth noting that when ϕ^l_{offset} in (9) is different for each robot, we can model the robot leg phases one by one instead of the solely nominal phase, and use a similar controller to synchronize them. However, the synchronization procedure may lead to unnatural contact profiles which are challenging to achieve on real robots.

C. MPC-based Locomotion Control

The robot legs have two different status: swinging in the air and stance on the ground. The state of the l-th leg can be determined by ϕ^l_{offset} and $\phi_{\text{threshold}}$, introduced in Section II-B, and then the desired gait is obtained. In the proposed framework, MPLC consists of a swing leg controller and a stance leg controller in terms of the leg states. The objective of MPLC is to track the desired trajectory from DMPFC and to achieve the generated gait from GSC.

1) Swing Controller: For a swing leg, the controller will first calculate the desired trajectory of the foot end in the air, which is based on the desired body velocity and trajectory. Then, the desired motor positions are calculated using inverse kinematics. Finally, a simple Proportional-Derivative (PD) controller is used to track the motor positions,

$$\tau_i = k_{p,i}(\theta_{d,i} - \theta_i) - k_{d,i}\dot{\theta}_i$$

where the subscript i implies the i-th motor, τ is the motor torque, θ_d the desired motor position, θ and $\dot{\theta}_i$ are the current motor position and velocity, respectively, and k_p and k_d are PD parameters, respectively.

2) Stance Controller: The stance controller computes the optimal ground reaction force for the stance leg, and then uses the Jacobian matrix J_i of the leg to obtain the motor torques:

$$\tau_i = J_i^\top f_i^*$$

where f_i^* is the ground reaction force of the i-th leg. An optimization-based MPC method is used to obtain f_i^*. Similar to [9], the dynamics model of the quadruped robot is linearized using the desired trajectory from DMPFC,

$$x_{i+1} = A_i x_i + B_i u_i,$$

where

$$\begin{aligned} &\forall i = t, \cdots, t+n-1, \\ &x = [\Theta^\top \quad p^\top \quad \omega^\top \quad \dot{p}^\top \quad -g]^\top, \\ &u = [f_1 \quad \cdots \quad f_4]^\top, \end{aligned}$$

$\Theta \in \mathbb{R}^3$ is the body orientation in the global frame, $p \in \mathbb{R}^3$ is the body position in the global frame, $\omega \in \mathbb{R}^3$ is the body orientation velocity in the body frame, $\dot{p} \in \mathbb{R}^3$ is the body linear velocity. Then, a QP is formulated to find the optimal ground reaction forces

$$\begin{aligned} \min \quad & \textstyle\sum_{i=t}^{t+n-1} (x_{i+1} - y_{i+1})^\top L_i (x_{i+1} - y_{i+1}) + u_i^\top K_i u_i, \\ \text{s.t.} \quad & x_{i+1} = A_i x_i + B_i u_i \\ & \underline{c}_i \leq C_i u_i \leq \bar{c}_i \\ & \forall i = t, \cdots, t+n-1, \end{aligned}$$

where n is the length of the predictive horizon, y_{i+1} is the desired states, L_i and K_i are the weighted diagonal

Fig. 4. Figure showing our quadruped robot Unitree A1 (right), and its simulation model in Pybullet [24] simulator (left). The UWB sensor and the onboard computer are circled by the blue and orange dotted lines, respectively.

matrices, $\underline{c}_i$, $\bar{c}_i$ and C_i are the coefficient vectors/matrix for the linearized friction constraints. In practice, we use $s_i^*(k+l|k)$ from DMPFC to construct y_{i+1}. The MPC control law at the time step t is defined as u_t^*, and then the optimal ground reaction force of the i-th leg can be obtained. Fast Convex MPC method [50] is used to solve the above optimization problem with a high update frequency for better control performance in trajectory tracking and gait stabilization.

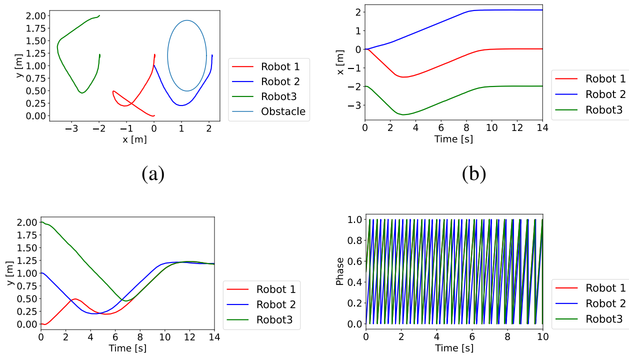
III. RESULTS AND DISCUSSION

In this section, the proposed control framework is validated in both simulation and the real world using three Unitree A1 robots [8], shown in Fig. 4. In real-world applications, the communication topology should be time-varying. In this paper, we fix the communication topology as $1 \leftrightarrow 2$, $2 \leftrightarrow 3$, which is a simple connected and undirected graph. For DMPFC, the controller setting is as follows. For all the robots, $\rho_i = 0.01$. For Robot 1 and 3, $\alpha_i = 1$ and $\beta_i = 0$, while $\alpha_2 = 0$ and $\beta_2 = 1$ for Robot 2. In this setting, Robot 1 and 3 only try to achieve the formation based on the relative distance from Robot 2, and thus do not require their absolute positions in the global frame. Moreover, Robot 2 tries to arrive at the desired position without considering the desired formation. The setting is similar to the leader-follower paradigm.

A. Simulation

The simulation experiment is conducted in PyBullet [24]. In this test, the initial value of the nominal phase of Robot 1 and 3 is 0.5, while that of Robot 2 is 0. The nominal stepping frequency in the gait model for the three robots is the same $f_0 = 2.4$. The initial positions are $(0,0)$, $(0,1.2)$, $(-2,2)$ for Robot 1, 2 and 3 respectively, and the desired positions are $(2,1.25)$, $(0,1.25)$, $(-2,1.25)$. The obstacle is placed at $(1.2,1.2)$.

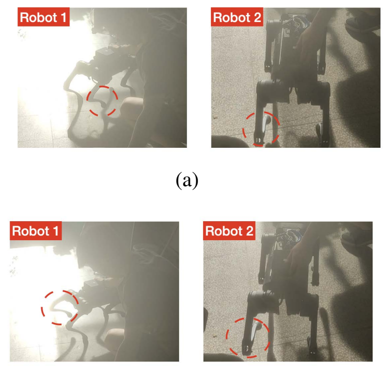
The test scenarios are shown in Fig. 1. It can be observed that Robot 2 has different leg states from Robot 1 and 3 at the beginning: the left rear and right front legs of Robot 2 are swinging while that of Robot 2 and 3 are on the ground. After several seconds, their leg states become the same, meaning that their gaits are synchronized. We can also observe that the three robots arrive at their desired positions and form a queue-like formation while avoiding collisions with the obstacle.



(a) (b)

(c) (d)

Fig. 5. Simulation Results. (a) The trajectory of the three robots in the global frame. (b) The trajectory in the x-axis. (c) The trajectory in the y-axis. (d) The phase trajectory.



(a)

(b)

Fig. 6. Gait synchronization of two real robots. DMPFC is off in this test. (a) Two robots have different initial phases at the beginning. (b) The contact schedule of the two robots is synchronized after seconds. The red dotted lines in (a) and (b) highlight one of the swing legs.

Fig. 5 shows the numerical results in detail. Robot 2 first bypasses the obstacle and then advances to the target position. Robot 1 and 3 do not move onto the desired position directly. Instead, they try to follow Robot 2 to achieve the desired formation, which is caused by our DMPFC setting. It can be also observed that the robots synchronize the gaits after 10 seconds, which is consistent with Theorem 1.

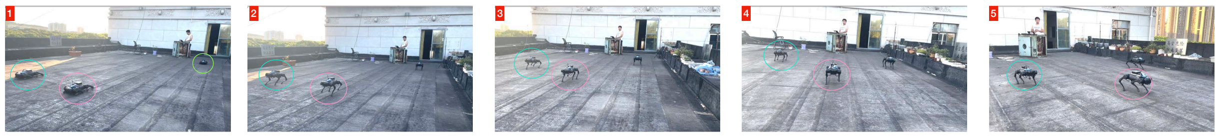
B. Real Robot Tests

Each robot is equipped with an onboard computer with Intel Core i7 2020 processor that runs Ubuntu Linux. Instead of a simultaneous localization and mapping (SLAM) module, we use a set of Ultra Wideband (UWB) sensors for self-localization and obstacle-localization. The hardware of the three robots is the same and can be visualized in Fig. 4.

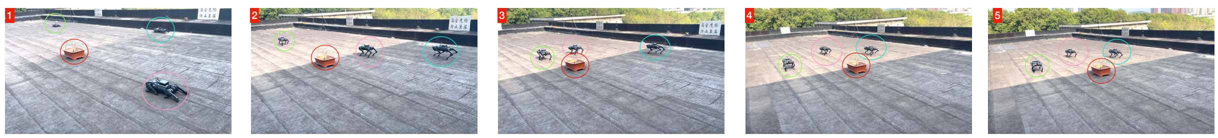
First, two robots are used to validate that GSC can achieve our significant objective, gait synchronization. The results are shown in Fig. 6. This concludes that GSC can be applied to real robots to synchronize the gaits. Next, we predefine three different shapes for formation control and randomly initialize the positions of the three robots to evaluate the overall control framework. The experimental results are consistent with that

(a)

(b)



(c)



(d)

Fig. 7. Experiments on real robots. Robot 1, 2, and 3 are circled in the green, pink, and blue colors, respectively. Four test scenarios are (a) Trot Abreast, (b) Queue, (c) Triangle Formation, and (d) Triangle Formation with Obstacle Avoidance. The red circle highlights the obstacle in the environment. In all the tests, the robots can achieve the predefined formation and meanwhile synchronize their gaits.

in the simulation as shown in Fig. 7. All the robots can arrive at the desired position to achieve the predefined formation and meanwhile synchronize their gaits. Moreover, each robot can avoid collisions with the other robots or obstacles in the environment. Please see our complementary video for the details.

Although the setting of all the controllers is the same, we find the control performance in the real robot tests is worse than that in simulation. Note that in Fig. 6, the phase of Robot 2 is consistent with that of Robot 1 within 1 second. However, we observe that the leg states of the two robots become identical after more than 1 second. The potential reason is that GSC is a synchronous control strategy over communication network and the time delays of wireless communication are not considered. Similarly, we also observe that the formation performance of the robots is worse than that in simulation. In addition to the time delays, another potential reason is that the accuracy of UWB is not satisfactory. Nevertheless, the experimental results show that the proposed framework is able to achieve our control objective on real robots reasonably well.

IV. Conclusion

This paper proposes a distributed control framework for multiple quadruped robots towards coordinated formation control and gait synchronization. We decompose the tasks and design three sub-controllers, i.e., DMPFC, GSC and MPLC, to achieve the overall objective. The experimental results show that the quadruped robot swarm is able to achieve the predefined formation, avoid the collisions and meanwhile synchronize the gaits. We believe our work extends the capability of quadruped robots towards more advanced swarm intelligence.

One limitation of our method is that DMPFC only computes the local optimum, which supposedly leads to inefficient coordination, or even failure. In addition, the time-delays are not considered in DMPFC or GSC. We leave the investigation of these challenges for the future work. Moreover, the lack of SLAM module limits the application of our method. In the near future, we plan to utilize the mature works from the SLAM community [51], [52] to navigate the quadruped robot swarm in wild and clustered environments.

Acknowledgment

This work was supported under the National Natural Science Foundation of China under Grant 62173155, Program for HUST Academic Frontier Youth Team, the CityU Start-Up Grant (Project No. 9610481), and Chow Sang Sang Group Research Fund sponsored by Chow Sang Sang Holdings International Limited (Project No. 9229076).

References

[1] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[2] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[3] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.

[4] H.-W. Park, P.-M. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[5] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[6] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[7] Boston Dynamics Spot, https://www.bostondynamics.com/products/spot.

[8] Unitree A1, https://www.unitree.com/products/a1/.

[9] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[10] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[11] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4730–4737.

[12] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, 2020.

[13] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, 2020.

[14] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[15] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[16] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning*. PMLR, 2022, pp. 773–783.

[17] D. Floreano and H. Lipson, "From individual robots to robot societies," p. eabk2787, 2021.

[18] H. Balta, J. Bedkowski, S. Govindaraj, K. Majek, P. Musialik, D. Serrano, K. Alexis, R. Siegwart, and G. De Cubber, "Integrated data management for a fleet of search-and-rescue robots," *Journal of Field Robotics*, vol. 34, no. 3, pp. 539–582, 2017.

[19] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," in *Field and service robotics*. Springer, 2014, pp. 33–47.

[20] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[21] H. Kolvenbach, G. Valsecchi, R. Grandia, A. Ruiz, F. Jenelten, and M. Hutter, "Tactile inspection of concrete deterioration in sewers with legged robots," in *12th Conference on Field and Service Robotics (FSR 2019)*. ETH Zurich, Robotic Systems Lab, 2019.

[22] I. D. Miller, F. Cladera, A. Cowley, S. S. Shivakumar, E. S. Lee, L. Jarin-Lipschitz, A. Bhat, N. Rodrigues, A. Zhou, A. Cohen, *et al.*, "Mine tunnel exploration using multiple quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2840–2847, 2020.

[23] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, *et al.*, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3306–3313.

[24] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[25] O. Feinerman, I. Pinkoviezky, A. Gelblum, E. Fonio, and N. S. Gov, "The physics of cooperative transport in groups of ants," *Nature Physics*, vol. 14, no. 7, pp. 683–693, 2018.

[26] T. D. Seeley and S. C. Buhrman, "Nest-site selection in honey bees: how well do swarms implement the" best-of-n" decision rule?" *Behavioral Ecology and Sociobiology*, vol. 49, no. 5, pp. 416–427, 2001.

[27] C. K. Hemelrijk and H. Hildenbrandt, "Schools of fish and flocks of birds: their shape and internal structure by self-organization," *Interface focus*, vol. 2, no. 6, pp. 726–737, 2012.

[28] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[29] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[30] F. Ducatelle, G. A. Di Caro, C. Pinciroli, and L. M. Gambardella, "Self-organized cooperation between robotic swarms," *Swarm Intelligence*, vol. 5, no. 2, pp. 73–96, 2011.

[31] J. S. Jaffe, P. J. Franks, P. L. Roberts, D. Mirza, C. Schurgers, R. Kastner, and A. Boch, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature communications*, vol. 8, no. 1, pp. 1–8, 2017.

[32] Y. Ozkan-Aydin and D. I. Goldman, "Self-reconfigurable multilegged robot swarms collectively accomplish challenging terradynamic tasks," *Science Robotics*, vol. 6, no. 56, p. eabf1628, 2021.

[33] Y. Ren, S. Sosnowski, and S. Hirche, "Fully distributed cooperation for networked uncertain mobile manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 984–1003, 2020.

[34] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots*, vol. 30, no. 4, pp. 385–397, 2011.

[35] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.

[36] N. Ayanian and V. Kumar, "Decentralized feedback controllers for multiagent teams in environments with obstacles," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 878–887, 2010.

[37] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[38] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[39] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[40] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017.

[41] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[42] P. Wang and B. Ding, "A synthesis approach of distributed model predictive control for homogeneous multi-agent system with collision avoidance," *International Journal of Control*, vol. 87, no. 1, pp. 52–63, 2014.

[43] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014.

[44] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[45] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[46] S. Xu, L. Zhu, and C. P. Ho, "Learning efficient and robust multi-modal quadruped locomotion: A hierarchical approach," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4649–4655.

[47] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," *arXiv preprint arXiv:2111.01674*, 2021.

[48] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[49] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[50] S. Xu, L. Dai, P. Cai, L. Zhu, and C. P. Ho, "Fcmpc: Fast convex model predictive control for quadruped robot," *arXiv preprint*, 2022.

[51] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[52] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *IEEE Transactions on Robotics*, 2022.