VaRSM: Versatile Autonomous Racquet Sports Machine

Fan Yang[†]
fanyang@deepcode.cc
DeepCode Robotics
China

Zhiwei Shi[†]
shizw@deepcode.cc
DeepCode Robotics
China

Sixian Ye
yesx@deepcode.cc
DeepCode Robotics
China

Jiazhong Qian
qianjz@deepcode.cc
DeepCode Robotics
China

Wenjie Wang
wangwj@deepcode.cc
DeepCode Robotics
China

Dong Xuan[‡]
dongx2012@sdu.edu.cn
Shandong University, China
The Ohio State University, USA

ABSTRACT

In this paper, we present a system called Versatile Autonomous Racquet Sports Machine (VaRSM in short). VaRSM can play table tennis, tennis and badminton with respective racquets. There are two major challenges in building VaRSM: first, VaRSM must be able to strike and return balls of variable speed and power on fields of different size with diverse racquet motions; second, VaRSM must track and predict balls' fast trajectories and move its body in extremely short intervals of time. To address these challenges, we design several innovative technologies, which we group into the physical hardware module and the control software module. In the physical hardware module, we create a high speed swerve-drive platform and a high-flexibility racquet arm, using configurable integrated drive units. In the control software module, we develop a proactive progressive control method that takes advantage of the hardware's physical capabilities to achieve early ball trajectory prediction, quick striking decision-making, and fast yet stable machine motion. We build a prototype system based on these technologies. Our experiments demonstrate VaRSM is able to strike and return table tennis, tennis and badminton balls with high success rates and is capable of playing with human players. To our knowledge, VaRSM is the first machine able to play three different ball games, and may hold great significance in education, research, economy, and society.

KEYWORDS

Sports machine, racquet arm, mobile platform, trajectory prediction, proactive progressive control

1 INTRODUCTION

We develop Versatile Autonomous Racquet Sports Machine (VaRSM), which can play three different ball games (table tennis, tennis, and badminton) with their respective racquets, as illustrated in Figure 1. This work has great significance in both research and practice. Racquet sports games place high demands on human player's strength, agility and reaction. We demonstrate that our design of mechanical architecture, electrical drivetrain and control strategies raise VaRSM's locomobility to match human athletic ability, using low-cost components and manufacturing. VaRSM has the potential to be commercialized as a training tool that can aid coaches and players alike. By involving advanced robotic technologies, it can garner interest in sports and robotics among the younger generations.

The development of VaRSM faces numerous difficulties. Table tennis, tennis and badminton are three very distinct activities, posing unique challenges related to the different ball characteristics, playing field size, and overall gameplay speeds. Even within one sport, such as table tennis, there are many challenges as incoming and outgoing balls vary in speed and spin. VaRSM must be able to intercept balls in most cases, and returning balls must land in targeted areas of opposite field. VaRSM must detect and track high-velocity objects, predict contact position, move the body and arm to intercept, then strike the ball, and finally reset for next strike within a short interval of time. In table tennis, tennis and badminton, the average ball travel intervals are 0.7, 1.6 and 1.0 seconds, respectively. VaRSM faces various challenges to fulfill the quick reaction requirement: 1) real-time prediction may be inaccurate and unstable inside the short time interval, especially in the beginning of the interval; 2) inaccurate and unstable prediction of ball trajectories may induce abrupt motion changes of VaRSM; 3) precise motion control becomes more difficult as VaRSM's motion speed increases.

To our knowledge, no other robotic system is able to play multiple racquet sports. Robots that are able to play only one game exist; for example, table tennis robots like the humanoid robot of Xiong *et al*[17], and Omron's parallel robot[9]. However, structural limitations prevent these robots from expanding to tennis and badminton. Stoev *et al* created a badminton robot with a slide rail to play badminton[14], however its speed and strength are insufficient for table tennis and tennis, and it is only capable of an overhead striking motion. VaRSM's unique physical and software design is versatile enough to adapt to three different racquet sports.

VaRSM has two major modules: one is the physical hardware module and the other is the control software module. In the physical module, we design a high-speed swerve-drive mobile platform, a 6-DoF (degree of freedom) racquet-swinging arm, and a stereo camera system. The racquet arm is installed on the mobile platform. The stereo camera system is separated from the mobile platform and the arm. All components used are commercially available and low-cost. We develop a dynamically configurable integrated drive unit (IDU) that combines an out-rotator motor and speed reducers into a single device to enhance power and torque density. IDU's design is compatible with a wide range of motor and speed reducer settings. IDUs are likewise driven by a standard driver circuit with configurable settings. The swerving unit's driving and steering

[†] The first two authors are co-primary authors.
[‡] Xuan conducted the work when he was on leave from the Department of Computer Science and Engineering at The Ohio State University, USA.

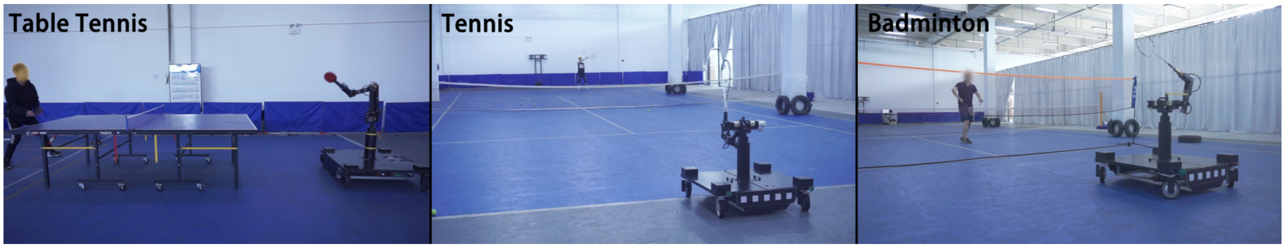


Figure 1: Game scenarios

components, as well as the racquet arm's six joints, are all driven by IDUs, with specialized mechanical and electrical setups. In the control module, we devise a proactive progressive control method. With this control method, VaRSM proactively conducts ball trajectory prediction, and quickly makes striking decisions and drives the mobile platform and the racquet to move, even at the very early stage of ball flight. Furthermore, the machine repeatedly performs the above actions in a very short time period (10ms) to progressively overcome the errors led by the pure proactive approach. As a result, our proactive progressive control method can achieve early ball trajectory prediction, quick striking decision making, and fast yet stable mobile platform and racquet arm motion.

To summarize, we have made the following contributions:

- We have created the world's first autonomous system that can play three **r**acquet sports (table tennis, tennis and badminton).
- We have developed novel mechanical architecture to strike and return balls, including a high-speed swerve-drive platform and a 6-DoF racquet-swinging arm, both with integrated drive units (IDU),
- We have devised a proactive progressive control method to meet both real-time and accuracy requirements of striking motion control.
- We have implemented a prototype and tested it in the real world. VaRSM can proficiently play table tennis, tennis and badminton . The average hit and return success ratios of these three games are 86.2%, 85.3% and 83.5%, respectively. VaRSM is capable of challenging human players in all three sports.

We place a short anonymised video on Youtube to demonstrate that VaRSM is able to play with human players in the three racquet sports (https://www.youtube.com/watch?v=LVDZn4ZX1sw&ab_channel=SportsMachine)..

2 SYSTEM DESIGN

2.1 Overview

VaRSM system framework includes the physical module and the control module, as depicted in Figure 2. The physical module consists of stereo cameras, a computer, a robotic arm holding the racquet (called racquet arm in this paper), a high-speed omnidirectional mobile platform, and configurable IDUs. Stereo cameras and the computer are placed on a man-height stand, several meters behind the machine. The control module consists of the perception component and the manipulation component. Stereo cameras provide high-fps images for the perception component to detect and localize both flying balls and the mobile platform, and predict both trajectories. After the intercept point is calculated, the manipulation component plans machine's motion to strike and return the ball. Motion commands are sent to the machine via wireless communication.

Controlling the machine to strike balls is a complex process. Control inputs include the ball's trajectory, machine's current location and velocity, and balls' return targets. Control outputs include the mobile platform's velocity, as well as angular position and speed commands for each joint of the robot arm.

We define the mobile platform's linear and angular velocities at time t as $V^p(t) := [v_x^p, v_y^p, \omega^p]$; the six joints' angular positions and speeds of the racquet arm at time t as $V^a(t) := [\theta_1^a, \cdots, \theta_6^a, \omega_1^a, \cdots, \omega_6^a]$; the mobile platform's pose (including its position and heading orientation) as $P^p(t)$; the ball's position as $P^b(t)$; the ball's trajectory from time 0 to t as $Traj^b(t) := [P^b(0), \cdots, P^b(t)]$; and the returned ball's target point on the opposite side of the field as P^b_{target}

The whole control process can be formulated as a system function $F_{control}$. At time t, velocity commands for the mobile platform $\hat{V}^p(t)$ and position/speed commands for racquet arm joints $\hat{V}^a(t)$ are obtained by Formula 1.

$$
\begin{aligned}
&[\hat{V}^p(t), \hat{V}^a(t)] = \\
&F_{control}(P^p(t), V^p(t), V^a(t), Traj^b(t), P^b_{target})
\end{aligned}
\tag{1}
$$

Here is the detailed control procedure: (1) at time t, the perception component measures $V^p(t)$ and $V^a(t)$ with on-board encoders, and get $P^p(t)$ and $Traj^b(t)$; (2) the perception component predicts the future trajectories of ball, the mobile platform and the racquet arm respectively, and calculate the probable intercept point; (3) the manipulation component proactively and progressively plans mobile platform and racquet arm's motion based on their status, the predicted intercept point on the machine's side of the field, and target point on the opposite side of the field P^b_{target}. (4) Motion commands are wirelessly sent to the platform and racquet arm.

To successfully return the ball to the target point, the machine must have high agility and reaction performance, which imposes high requirements of all the wheels' and joints' acceleration. Meanwhile, movements must be smooth and stable, without slippage and vibration, and the racquet arm must be able to perform diverse swinging motions consistently. Real-time communication is able to transmit control commands and feedback the machine's status

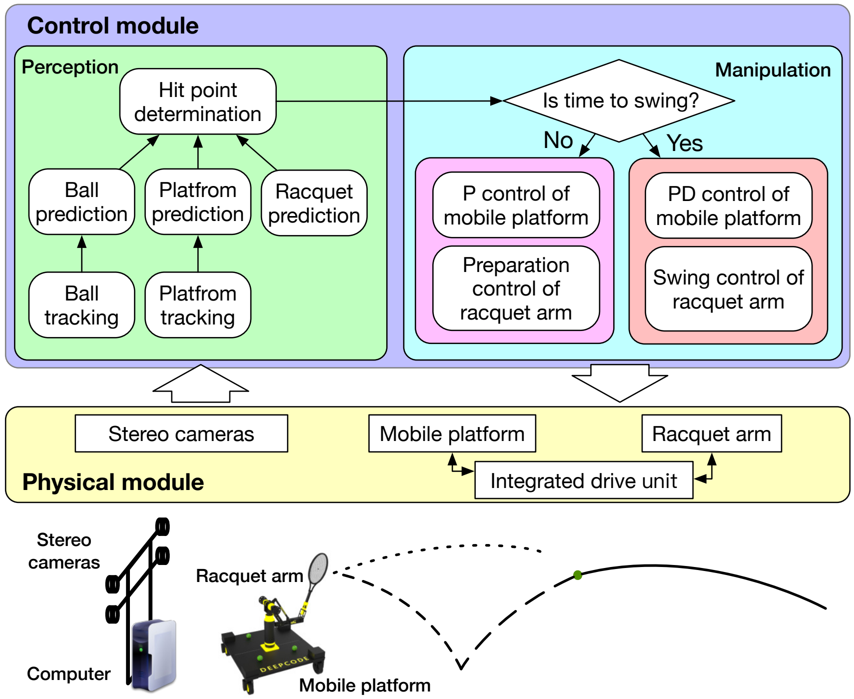


Figure 2: System framework

with low latency. Details of the physical module are presented in Section 2.2.

In addition, the control module must detect the ball, predict trajectories and generate control commands as early as possible. The progressive control method is used to mitigate prediction and control errors to make the machine's movement smooth and stable. Details for the control module are presented in Section 2.3.

2.2 Physical Module

We devise a light-weight mobile platform with the same speed and acceleration as an adult, as well as a racquet arm with small inertia capable of striking balls with sufficient power. We design integrated drive units with configurable hardware and software for wheels and joints to achieve the desired performances.

2.2.1 ***Integrated Drive Unit (IDU).*** The design of a configurable integrated drive unit with control solutions has the following two major benefits: (1) the design makes the the drive unit to have high power density and ease of integration due to the integration of the motor and speed reducer; (2) the design allows the drive unit to implement multiple drive modes via different parameter configurations, achieving high dynamics and flexibility.

External rotor brushless DC (BLDC) motors have been employed as power sources in various applications, such as scooters, UAVs and mechanical dogs. We are motivated by the motors' design advantages in terms of light weight and high torque. IDU is an integration of a BLDC motor and a speed reducer, in which the reducer is "implanted" inside the motor's stator. We use parallel shaft gear reducer, planetary gear reducer and cycloid reducer for different speed, torque, space and weight requirements. Compared with traditional servo system's external mounting of speed reducer on motor, IDU has the advantages of small size, light weight, high power density, and easy installation, which are critical for fast-moving robots. In VaRSM, IDU is applied in the mobile platform's swerve drive modules as well as the racquet arm's joint modules. IDU's power, gear ratio, speed, size, weight, and other parameters for each drive unit is designed based on the maximum performance requirement among various ball stroke motions, so that VaRSM can play various ball games.

In VaRSM, multiple components, such as the mobile platform and the racquet arm, are involved to complete a stroke motion. They impose different requirements to IDU characteristics. For example, the mobile platform requires the wheels' IDUs to achieve high speed and fast steering, and the racquet arm requires the joints' IDUs to control swing direction and provide stroke speed. Moreover, different stroke motions require different IDU parameters, which need to be adjusted in real time during a stroke. Therefore, we design a universal IDU driver to support various IDU configurations and working conditions. This makes IDU highly versatile and able to be used in every wheel and joint on the machine. Furthermore, the universal driver's control parameters can be adjusted both aforehand and dynamically, so that each IDU can adapt to different working conditions and achieve various stroke motions.

The IDU driver's design scheme is shown in Figure 3. It is composed of the user interface layer, the hardware abstraction layer and the motor drive layer. The user interface layer is implemented with bus control protocols and host computer interface. The hardware abstraction layer is used to receive and analyze control parameters

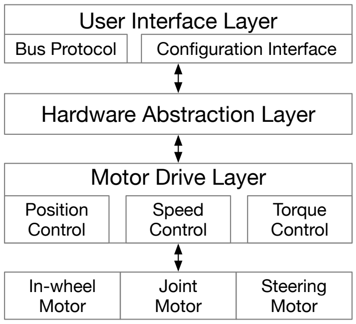


Figure 3: IDU driver design architecture

Sport	$V_{max}(m/s)$	$M(g)$	$D(cm)$	$P_{max}(mJ)$
Table tennis	32.0	2.5	4.0	80
Tennis	73.0	55.0	6.5	4015
Badminton	137	5.0	6	685

Table 1: Racquet sports balls parameters

from the user interface layer, generate drive control commands, and send them to the motor drive layer. A variety of drive algorithms are implemented in the motor drive layer to drive BLDC motors.

The universal driver's configurability benefits the machine's motion. Firstly, the universal driver can be configured into different modes in order to drive each individual IDU in the mobile platform's wheels and the racquet arm's joints for different purposes, such as moving and steering the platform, or swinging a racquet with a specific direction and speed. Secondly, as shown in Table 1, different balls vary in weight, size, maximum speed and power, hence the racquet's swing speed range must be adjusted correspondingly. According to momentum and mechanical energy conservation, there is a huge impact force during the contact of ball and racquet. Joint IDUs must be able to provide instantaneous power to keep the racquet rigid during the impact. IDU's control parameters, such as acceleration curve and PID coefficients can be adjusted dynamically, so that the racquet can get enough power at the hit point. Finally, the machine may play with human players of different levels, at which ball's speed and strength are varied. Different striking speeds and holding torques are required. According to incoming and outgoing ball's speeds, the control algorithm dynamically adjusts the internal parameters of the drivers to reduce the speed control's overshoot and increase the position control's holding torque.

2.2.2 ***Mobile Platform and Racquet Arm.*** To strike a ball, the machine needs two characteristics: "quick moving" and "quick swing". The machine's quick moving requires high acceleration and stable manipulation without slipping or rollover. To perform diverse stroke motions, the quick swing process needs to dynamically change the racquet's posture and speed near the hit point to simulate human hands.

Mobile Platform. For "quick moving", we create a four-wheel omni-directional high-speed mobile platform using IDU. Its diagram is shown in Figure 4. We use a light-weight aluminum alloy frame construction with eight IDUs to make four swerve wheels that provide high speed (by in-wheel IDU) and swift steering (by steering IDU). The rudder wheel incorporates an IDU with a BLDC motor with inside single-stage planetary gears, as well as a driver circuit.

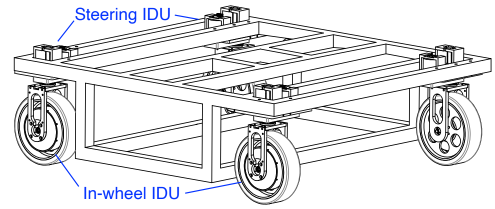


Figure 4: Mobile platform diagram

This integration minimizes the module's size and weight, and improves power and torque density. The high torque and acceleration enable the machine to move to the hit point within one second in most cases. We use rubber tires with high friction coefficients, which have stronger grip than Mecacnum and omni-directional wheels, accommodative to different kinds of field surface material. We decrease the height of the mobile platform and place the battery and control hardware at the bottom of the mobile platform to lower the machine's center of gravity, and the four-wheel independent drive mode enhances the swing process's stability. The mobile platform's IDU can be configured in real time to achieve various linear and angular movements of the machine. This can reduce the lateral force during a swing and avoid slipping and rollover.

Racquet Arm. For "quick swinging", we develop a light-weight racquet arm with high dynamics. We apply IDUs to the arm joints and build a 6-DoF racquet arm, so that the racquet can move precisely along planned path through the hit point in 3D space. The racquet arm's structure is shown in Figure 5. Considering stroke motions have various requirements for individual joint, including speed and torque, we select different outer rotor BLDC motors, reducers, and integration methods for each joint. Joint 1(J1), for example, is driven by an IDU with a BLDC motor and planetary gears because they need to provide swing speed without high position accuracy requirements; Joint 2(J2) and Joint 3(J3) are driven by IDUs with a BLDC motor and a harmonic reducer; Joint 4(J4), Joint 5(J5) and Joint 6(J6)'s IDUs incorporate direct-coupled planetary-cycloid hybrid reducers because of space and weight constraints.

Table tennis, tennis, and badminton all require six joints to act in coordination in order to implement stroke motions. However, each joint has a distinct function, just like human arm joints. For example, when making a flat stroke in tennis, J1 moves back and forth to offer the swing speed of the racquet, like human waist. J2 and J3 move up and down to provide an upward or downward tangential speed for the racquet and control swing height, like human shoulder and elbow. J4, J5 and J6 control the racquet's posture like human wrist. A variety of ball hitting movements can be made by configuring the motion parameters of these six joints.

The parameters of links connecting joints are derived based on the needs of various strokes. Links' lengths are determined according to human arm. This design enables the racquet arm to perform stroke motions like humans. The detailed length of each link can be found in Section 3.

2.3 Stroke Control

2.3.1 ***Control Strategy.*** Several problems exist in stroke control: Firstly, errors are introduced by ball trajectory prediction, mobile platform and racquet arm control and prediction. The predicted

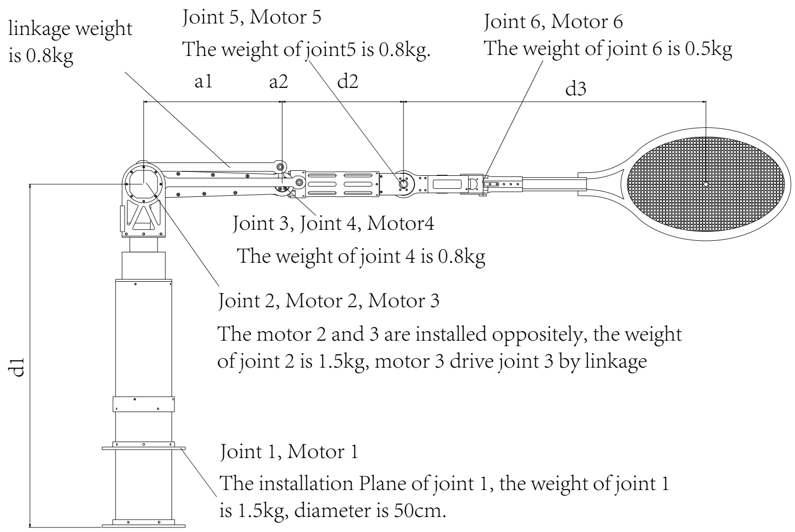


Figure 5: Racquet arm diagram

intercept point is determined by the cross point of the ball's and the racquet's trajectory predictions, which are obtained based on aerodynamics Extended Kalman Filter method (please refer to Appendix A for details) and mechanical motion models, respectively. These prediction errors accumulate and can cause stroke failure. Secondly, balls fly fast in the air and VaRSM must move across the field in a short period of time. When prediction is erroneous and unstable, movement commands may fluctuate. Despite we have improved VaRSM's mechanical structure and dynamic performance, control commands must be filtered to mitigate fluctuation. Finally, multiple stroke types for different ball games require varied motion of arm's joints. For example, for flat stroke in table tennis and tennis, J1 and J2 provide the racquet's speed, and others adjust the racquet's pose; while for push stroke in table tennis, J4 and J5 provides the racquet's speed, and others adjust the racquet's pose.

With regard to the first problem, VaRSM cannot wait for prediction's complete convergence before moving. Ball trajectory prediction gets more accurate as the ball flies. Prediction is more reliable during the falling than that during the rising stage before bouncing, and becomes more accurate after bouncing. Thus, we design a two-stage control strategy. At the first stage, when prediction is not accurate, the mobile platform should move to a region close enough to the predicted intercept point, and the racquet arm should get ready for swing. At the second stage, when prediction is accurate enough, the mobile platform moves to the exact intercept point, and the racquet arm begins to swing. Because we just move to an approximate area at the first stage, prediction errors are tolerated, and the moving commands fluctuate within a narrow range, which solves the second problem. For the final problem, we carefully design several types of stroke motions for different ball games, including drive and push for table tennis, ground stroke for tennis, overhead and underarm for badminton. In different stroke motions, arm joints are controlled in different modes.

2.3.2 *Control Algorithm*.

We design a proactive and progressive control algorithm based on the above basic ideas. The algorithm is shown in Algorithm 1. In the following, we first describe the algorithm workflow, and then present an important fine adjustment scheme for the racquet to hit the ball.

Algorithm Description. The algorithm runs as loops. Within each loop, ball and racquet trajectory predictions are conducted first, using aerodynamic models and computer vision technologies (Please refer to Appendix A for details of the trajectory predictions). The intercept point is estimated based on the predicted trajectories. Once the hit point is available. a proactive and progressive control method is applied to the mobile platform and the racquet arm to hit the ball.

The control of the mobile platform is split into two stages. At the first stage, the mobile platform is controlled to move to a rectangular approximate region of the intercept point before the ball is close enough for the racquet to start a stroke. The size of the rectangular region is shrunken as the ball approaches. We use P-control for platform's velocity, as shown in Formula 2. $\hat{V}^p$ denotes platform's velocity command, k_p is the proportionality coefficient, and $D^p_{hit\_zone}$ denotes platform's distance to it's closest point in the proximate zone.

$$\hat{V}^p = k_p \times D^p_{hit\_zone} \tag{2}$$

At the second stage, predictions become more accurate while the ball flies in the air, and we use PD-control for the platform to approach hit point, as shown in Formula 3. $D^p_{hit\_point}$ and V^p denote the platform's distance to hit point and it's current velocity towards hit point, respectively. k_d is the derivative coefficient. In this way, the platform can decelerate before arriving at the hit point.

$$\hat{V}^p = k_p \times D^p_{hit\_point} - k_d \times V^p \tag{3}$$

The racquet arm joints' control is also split into two stages. At the first stage, the joints are commanded to their preparation positions. At the second stage, the joints conduct stroke motions.

Algorithm 1 Proactive and Progressive Control Algorithm

```
Initialization: ball_traj = null, rac_traj = null, swing_flag = false, hit_point = null
while true do
  if BallDetected() then
    ball_traj = PredictBallTrajectory()
  end if
  if PlatformDetected() then
    rac_traj = PredictRacquetTrajectory()
  end if
  hit_point = PredictHitPoint(ball_traj, rac_traj)
  if hit_point is null then
    continue
  end if
  /* Control of the mobile platform */
  if not swing_flag then
    hit_zone = ProximityZone(hit_point)
    P_ControlOfPlatform(hit_zone)
  else
    PD_ControlOfPlatform(hit_point)
  end if
  /* Control of the racquet arm */
  if not swing_flag then
    RacquetArmInverseKinematics(racquet_hit_pose)
    PowerJointPrepareAngle()
    if current_time > power_joint_swing_time then
      swing_flag = true
    end if
    PowerJointsSpinToPrepareAngle()
    AdjustJointsSpinToHitAngle()
  else
    SpeedControlOfPowerJoints()
    PredictAdjustJointsHitAngle()
    PositionControlOfAdjustJoints()
  end if
end while
```

There is another point to mention. The racquet arm's six joints are insufficient to implement 6-DoF pose and 3-DoF velocity of the racquet at the same time. This means we may not be able to get a solution of the inverse kinematics of pose and velocity. Thus we design an approximate control method for the racquet arm.

The six joints are classified into two categories: power joints (P-joints) and adjustment joints (A-joints). P-joints provide racquet's speed and A-joints adjust some of the degrees of racquet's pose. For example, in table tennis's flat stroke, J1 and J2 are used as P-joints and J3,J4,J5,J6 are used as A-joints. In table tennis's push stroke, J4 and J5 are used as P-joints and J1,J2,J3,J6 are used as A-joints.

At the first stage, all joints are controlled to their preparation positions. At the second stage, due to prediction's unstability, joints' positions at the hit point are changing during the swing motion. P-joints don't respond to prediction changes, and keep spinning with a fixed speed. A-joints deal with the prediction changes and adjust the racquet's pose to hit the ball.

For the overall control frequency, because the stereo cameras are working at $100fps$, all the predictions and control commands are updated every $10ms$. The control algorithm is progressively performed every $10ms$ until the racquet hits the ball.

In summary, the core of the above algorithm is the proactive and progressive control method. For "proactive control", we mean that the control is applied before the predictions get completely accurate. For "progressive control", we mean that as the ball flies in air, we get more confident of the predictions, and finer control methods are applied to guide the racquet to hit the ball.

Fine Adjustment of Racquet. The final adjustment of A-joints is the key to a successful stroke. This is achieved by predicting the platform's trajectory and P-joints' angular positions.

Platform's trajectory is predicted with kinematics model, estimated dynamics model and control model. Kinematics is just the linear acceleration model (Formula 4). Here, we need to determine the platform's acceleration at time t, i.e. $a^p(t)$. We use a simplified dynamics model of swerve drive for $a^p(t)$: if the velocity command's magnitude is greater than current velocity's magnitude, then the platform accelerates with a fixed acceleration a^p, otherwise it decelerates with a fixed deceleration $-a^p$ (Formula 5). Velocity's direction is spinning towards command's direction with a fixed angular speed, which is the rotation speed of the swerve motors.

$$\begin{cases} P^p(t+dt) = P^p(t) + V^p(t) \times dt \\ V^p(t+dt) = V^p(t) + a^p(t) \times dt \end{cases} \tag{4}$$

$$a^p(t) = \begin{cases} a^p & |\hat{V}^p(t)| > |V^p(t)| \\ -a^p & o.w. \end{cases} \tag{5}$$

Now the problem is how to get velocity command $\hat{V}^(t)$. Here we use the P-control and PD-control in Algorithm 1 to estimate the velocity commands. The estimation is made based on the platform's positions and velocities iterated in Formula 4 and the ball trajectory predictions.

Arm joints' angular positions are predicted to estimate racquet trajectory. For P-joints, we use linear acceleration model to estimate their positions since they are in the speed control mode. A-joints are in the position control mode. We assume that they can rigidly follow the command positions. We can get the racquet's relative pose to the platform $POSE_p^r$ with 6-DoF arm's kinematics and the six joints' position predictions. Then the racquet's relative pose to the ground $POSE_g^r$ is obtained with $POSE_p^r$ and $POSE_g^p$(platform's relative pose to the ground).

With the control of platform and P-joints, racquet's estimated $POSE_g^r$ can be a little diverged from the target pose. A-joints need to be finely adjusted for the racquet to reach the target pose. The number of degrees that A-joints can adjust is no more than the number of A-joints. We firstly choose the most important ones from all the 6 degrees of racquet pose, and assign one A-joint to be responsible for each important degree. For example, in table tennis's drive stroke, J4 is responsible for racquet's yaw angle, J3 and J5 are responsible for z and y coordinates, respectively, and J6 is responsible for racquet's pitch angle.

Obviously, an A-joint will affect not only its assigned degree of pose, but also other degrees. Nevertheless, the assigned degree changes much more significantly with the corresponding A-joint than the others. Thus, we develop an iterative adjustment method for A-joints. Taking table tennis's drive stroke as example, we conduct the following adjustment procedures:

(1) given target pose's yaw angle, and other joints' estimated angular position at the hit point, we calculate J4's position.

(2) given target pose's z coordinate, and other joints' estimated angular position at the hit point, we calculate J3's position.

(3) given target pose's y coordinate, and other joints' estimated angular position at the hit point, we calculate J5's position.

(4) given target pose's pitch angle, and other joints' estimated angular position at the hit point, we calculate J6's position.

(5) iterate from 1) to 4) until convergence, i.e. results from adjacent iterations are close enough.

With the fine adjustment of racquet, final errors of prediction and control are mitigated and VaRSM can hit ball with high probability.

3 IMPLEMENTATION

We have implemented VaRSM based on the design presented in Section 2. VaRSM consists of the physical hardware module and the control software module.

Physical Module. This module has three main components: stereo cameras, a main computer, a high-speed omni-directional mobile platform, and a robust 6-DoF racquet arm. The racquet arm is installed on the mobile platform, as shown in Figure 6(b)(c), and their total mass $m = 40kg$. The stereo cameras and the main computer are placed behind them.

We use four 100-fps industrial cameras to construct a stereo vision system. The cameras' resolution is 1920x1200. Two of them are equipped with 8mm fixed-focus lens and the other two are equipped with 4mm lens. The cameras' views are able to cover the entirety of the play court. The main computer computer has an Intel i7 processor, 8G memory, and a GTX1080 graphics card. The four cameras' image captures are synchronized by an external trigger signal. The image processing time is 8-9ms for every four images.

For the mobile platform, we set the IDUs' parameters to match human running speed. The mobile platform needs to achieve the maximum speed $V = 5m/s$. The static friction coefficient between the rubber wheel and the ground was determined with testing, and is approximately $\mu = 0.8$, with a wheel diameter of $D = 150mm$. Ideally, the maximum acceleration of the mobile platform is $a_{max} = \mu g = 8m/s^2$. As such, the maximum static friction must be $F = \mu mg = 320N$. The torque provided by the four driving wheels must be greater than $T = \frac{F \times D}{2} = 24Nm$. A single swerve wheel needs to provide torque greater than $6Nm$. The maximum wheel spinning speed should be $N = \frac{V}{\pi D} = 10rps = 600rpm$. So the required motor power is $P = \frac{TN}{9550} = 0.38kw$. With similar calculation, we determine the parameters of steering IDU of swerve drive. The overall frame is made of 7−series aluminum. The in-wheel IDU motor's model is 8110, which is an outer rotor brushless motor and has a rated power of $600W$. The steering IDU motor's model is 3528, with a rated power of $170W$. In addition, the control module needs to know the mobile platform's position. Our method is to place a black and white visual label on the rear side of the mobile platform, and use the computer vision system to track it.

For the racquet arm, its mechanical settings are depicted in Figure 5, and its kinematics parameters are presented in the DH table

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	$80cm$	90°
2	0	−90°	0	0°
3	$40cm$	0°	0	90°
4	0	90°	$30cm$	180°
5	0	90°	0	0°
6	0	−90°	0	0°
7	0	0°	$35cm$	0°

Table 2: DH (Denavit and Hartenberg) table

Joint	Power(kw)	Gear ratio	Speed(rpm)	Torque(Nm)
1	3.5	25	250	150
2	2.3	50	140	150
3	2.3	28	250	100
4	0.4	33	120	40
5	0.4	33	120	40
6	0.2	19.2	550	4

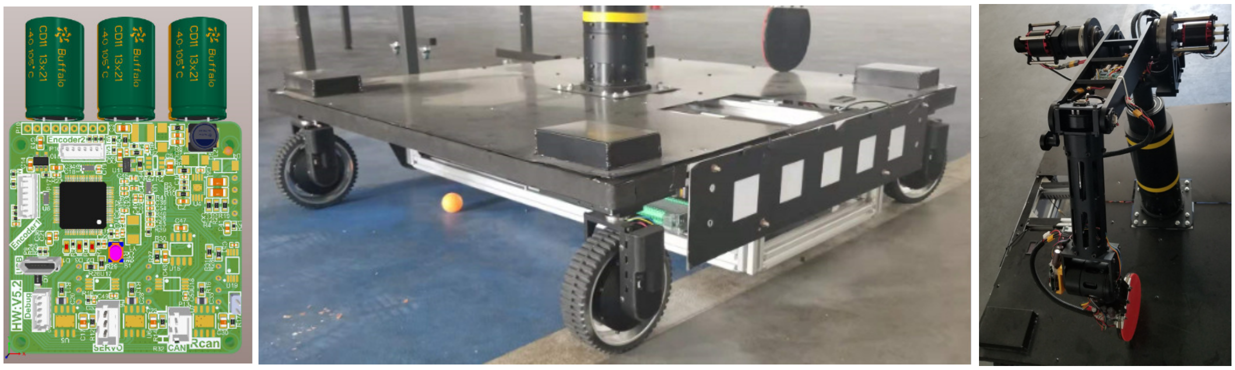
Table 3: Racquet arm joint parameters

(Table 2), where α_{i-1}, a_{i-1}, d_i and θ_i represent link twist, link length, link offset and initial joint angle, respectively. After the dimensions of the racquet arm are determined, the IDU parameters of each joint need to be computed according to the maximum speed and torque requirements of each joint. For example, the speed and torque requirements of J1 for a tennis flat stroke are the greatest among all stroke motions. If the racquet center point's speed reaches $10m/s$ in a flat stroke, and if the arm length is $1m$, then the angular speed requirement for J1 is $10rad/s$. If the acceleration space for J1 is $\frac{\pi}{4}rad$, the angular acceleration is $\frac{200}{\pi}rad/s^2$. Considering the total mass of the racquet arm and the held tennis racquet is $4kg$, the estimated load torque is $4kgm^2$, the required torque is $\frac{800}{\pi} \approx 255Nm$, and the effective power is approximately $2.55kw$. IDU specifications for racquet arm joints are listed in Table 3.

The IDU driver is implemented with an ST STM32F405RGT6 single-chip microcomputer, with $168Mhz$ operating frequency, as shown in Figure 6(a). The main control unit of the mobile platform and the racquet arm is an ST STM32H750VBT6 single-chip microcomputer, with $400Mhz$ operating frequency, and dual-channel 1Mbps CAN bus for separately controlling the mobile platform and the racquet arm.

All three components are controlled by a main computer. The main computer and the stereo cameras are connected with a USB cables. The main computer controls the mobile platform and the racquet arm via radio frequency. Specifically, we have developed a high-speed wireless protocol to ensures the real-time communication capability. Please refer Appendix B for details.

Control Module. The control module is implemented in C++ on Ubuntu 16.04. It includes four threads: the computer vision thread, the trajectory prediction thread, the machine control thread, and the stroke strategy thread. The computer vision thread handles four pictures every $10ms$ for detection and localization of ball and the mobile platform. The trajectory prediction thread tracks and predict balls' trajectory. If a new trajectory is found, the stroke strategy thread determines the speed and spin of the incoming ball. If the predicted trajectory is updated, the machine control thread generates and sends speed and position commands to the mobile platform and the racquet arm.



(a) IDU driver (b) Mobile platform (c) Racquet arm

Figure 6: Hardware implementation

Sports	Average	Stdev	Max	Min
Table tennis	456.93	55.72	632	315
Tennis	1645.36	103.06	1883	1332
Badminton	1125.85	202.41	1498	584

Table 4: Execution Time of Mobile Platform

Sports	Average(ms)	Stdev(ms)	Max(ms)	Min(ms)
Table tennis	408.13	25.96	438	282
Tennis	321.89	29.61	562	273
Badminton	302.33	10.76	372	291

Table 5: Execution time of Racquet Arm

4 EXPERIMENTS

In this section, we present our experimental evaluation results of VaRSM. Our experimental evaluation includes three racquet sports: table tennis, tennis and badminton. We test VaRSM's performance with two methods: returning balls fired from a automatic ball launcher, and and playing with human opponents. The number of trial shots with the ball launching machine are: 1410 for table tennis, 211 for tennis, and 225 for badminton. For each shot, we record the execution time interval of the mobile platform and the racquet arm, the ball's trajectory prediction error with time, and the success or failure of the return.

Execution Time of the Mobile Platform and the Racquet Arm. Table 4 and 5 list the execution time of the mobile platform and the racquet arm respectively. The execution time of the mobile platform is the time interval starting when the machine first decides to start moving, and ending when the machine strikes the ball. The execution time of the racquet arm is the time interval starting at the moment the arm starts to move, ending at the moment of striking the ball.

In order for the system to be able to intercept incoming balls, the execution time must be less than the ball travel time. In the 1410 table tennis balls trials, the average travel time is 456.93*ms*. The average execution time of the racquet arm playing table tennis is 408.13*ms*, The situation for tennis strokes is similar. As shown in Table 4 and Table 5, the execution time of the racquet arm playing tennis is 321.89*ms*, which is much less than the average flight time of tennis. The time required for a badminton racquet to hit the ball is 302.33*ms*. From the above data, we can see that the racquet arm is fast enough to intercept balls in all the three sports, compared with their average ball flight intervals of 0.7, 1.6 and 1.0 seconds.

The average execution times of the mobile platform in table tennis, tennis and badminton experiments are 456.93*ms*, 1645.36*ms*, and 1125.85*ms*, respectively. They are less than the average ball travel intervals of the respective sports.

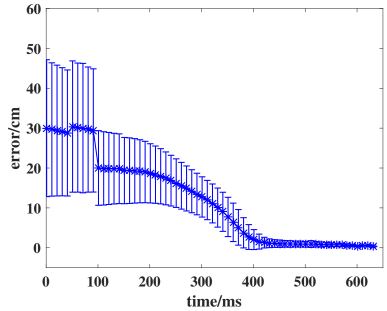
Item	top spin			no spin			back spin		
S#	162	155	164	168	160	153	140	150	158
R#	154	139	146	135	133	123	119	126	140
RR	91.27%			81.29%			85.94%		
OR	86.17%								

Table 6: Successful return rate of table tennis

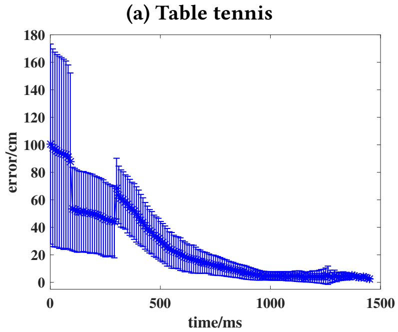
Trajectory Prediction Error. The trajectory prediction error is defined as follows: the deviation between the predicted position of the ball at the moment of striking and the actual striking position. During the flight of the ball, the predicted trajectory will be continuously updated and the prediction error will be also continuously calculated as the number of detected ball positions increases. The prediction errors of different racquet sports are shown in Figure 7. From the trajectory prediction error statistics, the trajectory prediction accuracy of table tennis in the last 250*ms*, tennis in the last 500*ms*, and badminton in the last 300*ms* are high enough for the racquet arm to adjust its speed and pose to make a successful stroke.

Successful Stroke Rate. The 1,410 table tennis ball launcher trials are divided into 9 groups, each group contains 140 to 170 trials. The statistics of the successful stroke rate are shown in Table 6. S#, R#, RR, OR denotes number of serves, number of successful returns, successful return rate of each group and overall successful return rate, respectively. It can be seen from Table 6 that the success rate of topspin ball is higher than the ones of backspin and non-spin balls. This is because strokes of backspin and non-spin balls require larger elevation angle of the racquet compared to topspin balls. This makes the racquet's effective stroke surface smaller, and increase misses and unstable strokes.

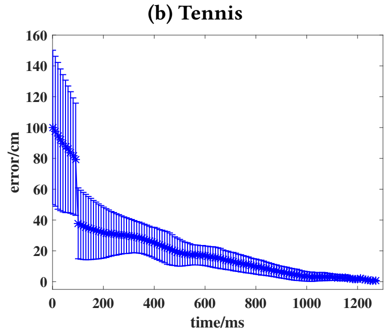
There are 211 tennis ball launcher trials, divided into 5 groups, each serving 40 to 46 trials. The statistics of the successful return rate are shown in Table 7.



(a) Table tennis



(b) Tennis



(c) Badminton

Figure 7: Hit point prediction error with time.

S#	42	46	40	40	46
R#	36	34	33	38	39
RR	85.7%	79.1%	82.5%	95%	84.8%
OR	85.3%				

Table 7: Successful return rate of tennis

S#	44	43	46	46	46
R#	36	38	41	39	34
RR	82%	88.4%	89.1%	84.8%	74%
OR	83.5%				

Table 8: Successful return rate of badminton

There are 225 badminton shuttlecock launcher trials, divided into 5 groups, each serving 43 to 46 trials. The statistics of the successful return rate are shown in Table 8.

These statistics demonstrate that the machine can return balls with high success rate.

In addition, we have conducted preliminary experimental tests on playing with human players. It is difficult to quantitatively evaluate the performance due to the inconsistency of incoming balls from human players. We place a short anonymised video on Youtube to demonstrate that VaRSM is capable of playing with human players in all the three ball games (https://www.youtube.com/watch?v=LVDZn4ZX1sw&ab_channel=SportsMachine).

5 RELATED WORK

In this chapter, we introduce and discuss the work related to our system, including robotic systems and related technologies, such as actuators, mobile platforms [10, 12], mechanical arms [4, 15, 17, 19] and hitting control technologies [16], etc.

In the past years, we have seen great efforts being made on controlling a machine to hit a flying object. To the best of our knowledge, no machine that can play multiple racquet games has been proposed. Nevertheless, there are some machines that can play one type of game, such as Xiong *et al*'s humanoid table tennis robot [18], Kyohei *et al*'s table tennis robot [9] and Stoev, J *et al*'s badminton robot[14]. Due to the limitations of their locomobility, they can hardly be extended to other ball games. To the best of our knowledge, there is no machine that can play tennis with people yet.

The mobile platform is one of the most important parts for a machine to move to ball contact position on time. Compared to legged structure, omni-directional wheel-drive platform is more suitable on flat ball fields. Mecanum wheel [10] is a good choice for omni-directional movement. Salih *et al* proposed design and control methods of a mobile platform with four mecanum wheels [11]. The Omni-wheel was first invented by J. Grabowiecki [10] and various improvements have been proposed thereafter. Sharbafi *et al* designed an omni-directional three-wheel robot [12]. Song *et al* used sensors to mitigate moving vibration of a four-omni-wheel robot [13]. Both mecanum and omni-wheels suffer from slippage and low power efficiency, and are inadequate for our requirements.

Serial and parallel robotic arms are widely used on industry and service robots. Cooperative robot arms have enough power, speed and precision to play table tennis. They are light-weighted, but still too heavy to be assemble on a fast-moving platform. Jia *et al* developed a batting system with 2-DoF robot arm fixed on a surface that can strike a slowly flying object to a target position [7]. Their research focused on the contact model of the robot arm and flying objects with on 2-DoF robot arm and 2D motions. Masaru *et al* proposed a novel control method for high-speed robot arm [16]. They designed a servo system with dynamic compensation algorithms in order to improve arm's stability in high-speed mode.

Motors are the power source of robots. Most mobile platforms and arms are driven by servo or brushless motors. Fisher *et al* proposed a high-precision servo system that works well in low-velocity mode [5]. Brushless motors are inexpensive but inaccurate, but they are reformed and used in some quadrupted robots in recent years [8]. This provides intuitive ideas for our IDU.

Machine learning and aerodynamics are two major methods in flying objects' trajectory prediction. Both methods utilize detected positions of a flying object to predict its future trajectory. Gomez *et al* proposed a trajectory prediction method using deep learning [6]. Specifically, they use encoding and decoding deep neural networks to do prediction. Online learning is also supported to mitigate error. Zhang *et al* proposed a trajectory prediction method based on computer vision(CV) detection and aerodynamics model[21]. 3D positions of the flying object are determined with CV technologies and aerodynamics is used to predict its afterwards positions. However, there is still much room for improvement of prediction's accuracy and real-time performance.

6 DISCUSSION

Our current design and implementation are still relatively preliminary. There is a lot of future work to complete.

Regarding the ball trajectory prediction [2, 6, 20], we may also consider to use a machine learning approach in addition to the aerodynamic model. Each of these two approaches has its advantages and disadvantages. The application of the aerodynamic model is relatively simple to implement. It can analyze specific parameters or mechanical problems based on the prediction results and make parameter adjustments. However, its debugging process is complicated and the accuracy of the ball rotation determination is not high. On the contrary, the machine learning approach is more complicated to implement, and it may be necessary to train the model specifically for different situations. However, it may have better prediction accuracy for some specific situations.

At present, the stereo cameras are fixed on the ground behind the mobile platform (and the racquet). The main advantage of this deployment is that the positioning accuracy of the ball and the mobile platform is high. Another option is to place the stereo cameras on the mobile platform, and the camera moves with the platform to strike the ball. Exploring this option is part of our future work.

Regarding the ball striking control [1, 3], our current solution has much room to improve. For example, the choice of the intercept point is not necessarily optimal. We plan to conduct a more in-depth analysis of this control problem for a better solution. Also, we can improve the actual striking process. Currently, due to various errors from multiple parties (e.g. mechanical and electrical ones etc.), it is not always possible to ensure that the racquet reaches the target posture. It has a relatively large impact on the accuracy of the return balls. As a part of our future work, we need to minimize the errors of all parties and improve the racquet arm control algorithm.

7 CONCLUSION

This paper introduced VaRSM, a machine that can autonomously play three racquet sports (table tennis, tennis and badminton). In VaRSM, we have designed a series of innovative technologies: high-speed and flexible omni-directional mobile platform and robotic arm based on IDUs; a proactive progressive control method to achieve early ball trajectory prediction, quick striking decision making, and fast yet stable mobile platform and racquet arm motion. We have implemented the system and conducted real-world experimental evaluation. Experimental data and videos have demonstrated VaRSM can proficiently play table tennis, tennis and badminton.

Our current implementation is still preliminary, and there is a lot of future work to be done. We plan to continuously improve this system, aiming for it to challenge even professional athletes.

ACKNOWLEDGEMENT

We sincerely thank anonymous reviewers and the area chair for their valuable comments. We acknowledge the support of Jingchuan Wang, Songquan Shi, Yiran Xuan, Haochen Wang and Huiyang Gao to our platform development, data collection and paper revision.

REFERENCES

[1] E. Brenner, J. Smeets, and M. Lussanet. 1998. Hitting moving targets. Continuous control of the acceleration of the hand on the basis of the target's velocity. *Experimental Brain Research* 122, 4 (1998), 467–474.

[2] L. J. Briggs. 1959. Effect of Spin and Speed on the Lateral Deflection (Curve) of a Baseball; and the Magnus Effect for Smooth Spheres. *American Journal of Physics* 27, 8 (1959), 589–596.

[3] B. Bäuml, T. Wimböck, and G. Hirzinger. 2010. Kinematically Optimal Catching a Flying Ball with a Hand-Arm-System. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[4] B. K. Cho, S. S. Park, and J. H. Oh. 2010. Controllers for running in the humanoid robot, HUBO. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*.

[5] D. S. Fisher, S. M. Keeler, and J. A. Mount. 2000. Data sector/servo split generator for a disk drive. (2000).

[6] S. Gomez-Gonzalez, S. Prokudin, B. Scholkopf, and J. Peters. 2020. Real Time Trajectory Prediction Using Deep Conditional Generative Models. *IEEE Robotics and Automation Letters* 5, 2 (2020), 970–976.

[7] Y. B. Jia, M. Gardner, and X. Mu. 2019. Batting an in-flight object to the target. *The International Journal of Robotics Research* 38, 4 (2019), 451–485.

[8] B. Katz, J. Di Carlo, and S. Kim. 2019. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 6295–6301.

[9] A. Kyohei, N. Masamune, and Y. Satoshi. 2020. The ping pong robot to return a ball precisely. *Omron TECHNICS* 51 (2020), 1–6.

[10] R. Rojas. 2006. A short history of omnidirectional wheels. *white paper* (2006).

[11] J. Salih, M. Rizon, S. Yaacob, A. H. Adom, and M. R. Mamat. 2006. Designing Omni-Directional Mobile Robot with Mecanum Wheel. *American Journal of Applied Sciences* 3, 5 (2006).

[12] M. A Sharbafi, C. Lucas, and R. Daneshvar. 2010. Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 6 (2010), 630–638.

[13] J. B. Song and K. S. Byun. 2010. Design and Control of a Four-Wheeled Omnidirectional Mobile Robot with Steerable Omnidirectional Wheels. *Journal of Robotic Systems* 21, 4 (2010), 193–208.

[14] J. Stoev, S. Gillijns, A. Bartic, and W. Symens. 2010. Badminton playing robot - a multidisciplinary test case in Mechatronics - ScienceDirect. In *Mechatronic Systems*. 725–731.

[15] T. Sugihara and Y. Nakamura. 2002. Whole-body cooperative balancing of humanoid robot using COG Jacobian. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*.

[16] M. Uchiyama. 2009. Formation of High-Speed Motion Pattern of a Mechanical Arm by Trial. *Transactions of the Society of Instrument and Control Engineers* 14, 6 (2009), 706–712.

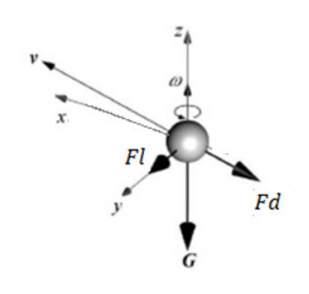
[17] R. Xiong, Y. Sun, Q. Zhu, J. Wu, and J. Chu. 2012. Impedance Control and its Effects on a Humanoid Robot Playing Table Tennis. *International Journal of Advanced Robotic Systems* (2012), 1.

[18] R. Xiong, Y. Sun, Q. Zhu, J. Wu, and J. Chu. 2012. Impedance control and its effects on a humanoid robot playing table tennis. *International Journal of Advanced Robotic Systems* 9, 5 (2012), 178.

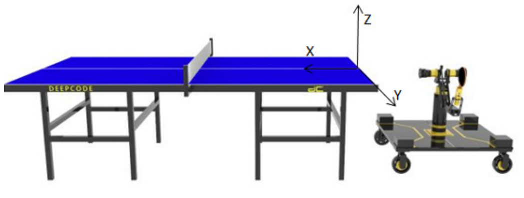
[19] E. Yoshida, M. Poirier, J. P. Laumond, O. Kanoun, and K. Yokoi. 2008. Whole-body motion planning for pivoting based manipulation by humanoids. In *2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA*.

[20] Y. Zhang, W. Wei, D. Yu, and C. Zhong. 2011. A tracking and predicting scheme for ping pong robot. *Journal of Zhejiang University Science C* (2011).

[21] Z. Zhang, D. Xu, and M. Tan. 2010. Visual Measurement and Prediction of Ball Trajectory for Table Tennis Robot. *IEEE Transactions on Instrumentation and Measurement* 59, 12 (2010), 3195–3205.



(a) Forces on ball



(b) Field coordinates

Figure 8: Flying ball force analysis

A BALL TRAJECTORY PREDICTION

VaRSM tracks and predicts the ball's trajectory in order to hit the ball. Extended Kalman Filter(EKF) is adopted to track the ball and estimate its latest state. Aerodynamic models are applied to predict its subsequent trajectory.

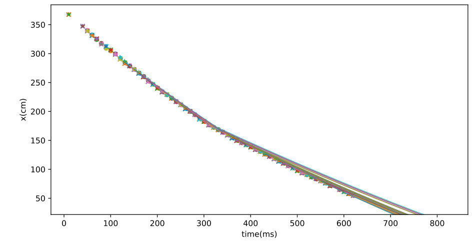
We define the state vector of EKF as $[x, y, z, v, \theta_v, \phi_v, \omega]^T$, representing ball's x,y,z coordinates, magnitude, yaw and pitch angle of linear velocity, and spinning speed. We only consider pure top- and back- spin, whose spin axis is horizontal and perpendicular to the ball's linear velocity. When there are 10 detected balls in the initial part of trajectory, EKF's initial state is obtained by polynomial fitting.

Aerodynamic prediction of ball's trajectory begins after EKF's initialization. To do so, we firstly make force analysis of the ball. Forces on the ball include gravity G, drag force F_d and Magnus force F_l. The forces imposed on the ball and ground coordinates are shown in Figure 8. Drag force Fd and Magnus force Fl at the nth ball in the trajectory can be calculated with Formula 6. f_d and f_l denote drag and Magnus force coefficients.

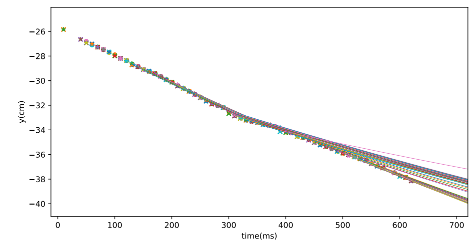
$$\begin{bmatrix} Fd_n \\ Fl_n \end{bmatrix} = \begin{bmatrix} 0.5 \times f_d \times v_n^2 \\ \frac{0.5 \times f_l \times v_n^2}{2 + v_n / w_n} \end{bmatrix} \tag{6}$$

With this force analysis, we derive linear velocity's and spinning speed's transfer functions from nth to $(n+1)th$ ball as shown in Formula 7, and position's transfer functions as Formula 8. m, g, dt denotes ball's mass, the acceleration of gravity, and time span between nth and $(n+1)th$ ball. cos and sin are represented by c and s for short. We predict the ball's trajectory with these formulas.

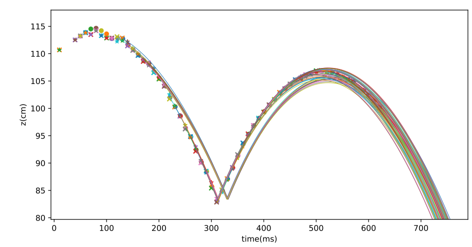
$$\begin{bmatrix} v_{n+1} \\ \theta_{n+1} \\ \phi_{n+1} \\ \omega_{n+1} \end{bmatrix} = \begin{bmatrix} v_n - (Fd_n/m + g \times s\theta_n) \times dt \\ \theta_n + (c\theta_n Fl_n/m - g \times c\theta_n)/v_n \times dt \\ \phi_n + (s\theta_n Fl_n/(m \times v_n) \times dt \\ \omega_n \end{bmatrix} \tag{7}$$



(a) x-time predictions



(b) y-time predictions



(c) z-time predictions

Figure 9: A trajectory prediction sample

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} c\theta_n c\phi_n & (-c\theta_n Fd_n - s\theta_n Fl_n) \times c\phi_n & x_n \\ c\theta_n s\phi_n & (-c\theta_n Fd_n - s\theta_n Fl_n) \times s\phi_n & y_n \\ s\theta_n & -s\theta_n Fd_n + c\theta_n Fl_n - mg & z_n \end{bmatrix} \begin{bmatrix} v_n dt \\ dt^2/2m \\ 1 \end{bmatrix} \tag{8}$$

When there are 20 detected balls in the trajectory, ball's spinning speed ω is estimated for better prediction. Ball's velocity v and acceleration a are updated with polynomial fitting, as well as drag force Fd and resultant force $F = ma$. Magnus force Fl is derived by deducting mg and Fd from F, and spinning speed ω is computed from Fl. 20 balls is sufficient for estimating ω, and this makes aerodynamic prediction more accurate.

Ball's trajectory prediction runs for each 1ms and bouncing is also considered. When the ball's predicted height is lower than the ground or table surface, a bouncing is triggered and the state vector after bouncing is updated.

A trajectory prediction sample's projection on xoz plain is shown in Figure 9. After the 10th point, the prediction begins. At the

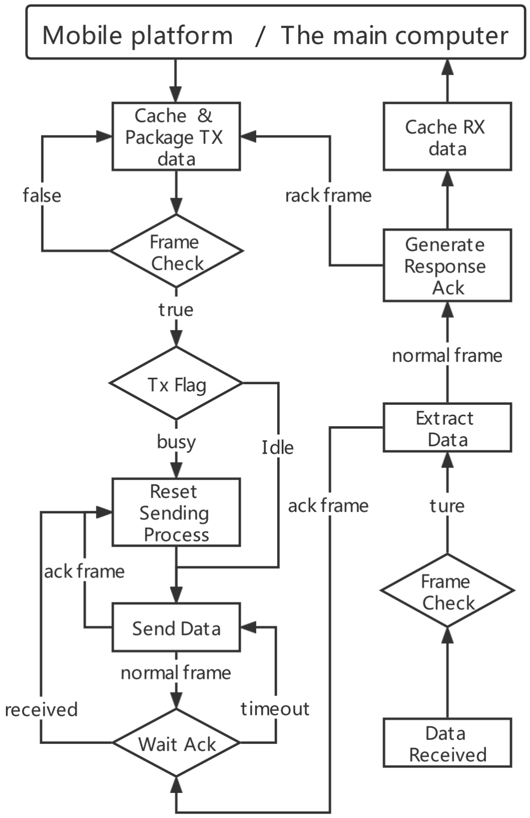


Figure 10: Communication flow chart

$20th$ point, the prediction becomes significantly better due to the calculation of ω.

B WIRELESS COMMUNICATION

We develop a high-throughput wireless protocol to ensure the capability of real-time communication between the main computer and the mobile platform with the racquet arm.

The main computer needs to control the robot's pose in real-time in order to strike the ball. The motion commands are sent to the robot at a fixed frequency of 100Hz. With high communication delay, the robot will easily miss the best interception point, or even exceed the flight time of the ball. With poor anti-interference capability, the robot will miss some wireless commands and cannot respond in real-time.

We design a high-speed self-correcting wireless communication system to realize real-time communication between the main computer and the robot. To achieve high reliability and low latency, we design a full-duplex communication scheme with different TX and RX channels, and develop a limited-time retransmission mechanism. We employ the STM32 USB interface to establish a 480Mbps channel to reduce the delay between the communication device and the main computer.

We use the SX1280 module based on LORA technology to realize wireless transmission and reception. LORA technology has forward error correction(FEC) capability, which can reduce package loss and retransmissions in a high-interference channel. Compared with traditional modulation schemes based on FSK or OOK, the LORA modem uses spread spectrum modulation to increase the range and robustness of the radio communication link. LORA modem's excellent anti-interference capability provides up to 19.5db co-channel suppression. This allows LORA communication to coexist with WiFi and Bluetooth networks in high spectrum usage scenarios. In order to ensure the reliability of data transmission, we design a retransmission mechanism with time constraints. A data frame is considered lost if no ACK is received for it. The lost frames are retransmitted only before the next frame is generated.

Figure 10 shows the communication flow between the main computer and the mobile platform. On the TX side, when the communication device receives a complete package from its host, the package is sent through the RF transceiver if the current TX flag is idle. At this time. the previous package will be discarded even if it's not transmitted successfully. When the transmission is completed, the device will wait for ACK package from the RX side. If no ACK is received within the specified time, the data package is retransmitted until the ACK is received or a new data package is ready to be sent. When the maximum number of retransmissions is exceeded, the current package is discarded. On the RX side, when a valid data package is received, an ACK package is sent back to the TX side. When a data package and an ACK package are needed to be transmitted at the same time, the two packages merge into a new one and are transmitted together, which improves real-time performance.