

Motion Plan Execution Using Empirical Data for an Omni-Directional Multi-Ball Drive Robot

Development and control of a second generation prototype for an all-terrain omni-directional mobility platform with spherical wheels

Akin Tatoglu, Biruk A. Gebre, Kishore Pochiraju
Design and Manufacturing Institute
Mechanical Engineering Dept.
Stevens Institute of Technology
Hoboken, New Jersey, United States

Abstract— A prototype for a mobile ground robot platform that makes use of spherical ball wheels for mobility was developed and presented in a prior paper. This multi-ball drive robot platform was prototyped with the aid of additive manufacturing technology (3D printing) and was shown to have agile omni-directional maneuvering capabilities. In this paper we present design improvements to the existing platform as well as an empirical methodology to measure system parameters and generate commands to execute a given motion plan. The design improvements include the addition of an articulated suspension system for each leg to improve maneuverability over various types of terrains and design modifications to the ball support structure to help improve motion accuracy and robustness. Prior to performing a full dynamic model of this complex system we examine the use of simplified kinematic model to execute motion command for a given path under open loop control. The objective is to generate an open loop control system that is able to execute motion commands for the vehicle similar to a skilled operator. The empirical modeling methodology presented utilizes iteratively generated experimental data to estimate kinematics variables and generate input motion commands for a low level controller. Mathematical modeling and experimental procedure and algorithms are presented in the paper. Preliminary test results verify that the proposed methodology is able to traverse a given path with a similar level of accuracy and speed of a skilled operator.

Keywords—*motion plan execution; empirical modeling; multi-ball drive; 3D printing; mobile ground robot*

I. INTRODUCTION

Mobile ground robots are being employed in a number of different application domains ranging from industrial application (logistics and warehouse automation) to public safety (disaster response and security) to military application (IED disposal) and commercial application (tele-presence and home cleaning robots). As such a number of ground mobility platforms have been developed to meet the specific needs of each application. These platforms range from wheeled and tracked vehicles to legged robots such as bipeds and quadrupeds [1]. In this paper we present a multi-ball drive robot that is part of a relatively new class of mobile ground robots platforms that utilize spherical or ball wheels [2-7]. The robot platform presented in this paper, named

“ATOM” (All Terrain Omnidirectional Mobility), is a second generation prototype that aims to extend upon the capabilities of the first generation presented in [5].

In [5] the basic concept of a multi-ball drive mobility platform was presented and some of the challenges involved in using spherical wheels were discussed. Solutions for many of these challenges were presented by designing components that leveraged the advantages of additive manufacturing technologies (3D printing). One of the main advantages leveraged was the fact that design complexity is free when fabricating components using additive manufacturing. It was therefore possible to design and fabricate intricate fully assembled components of various sizes using 3D printing that would have been prohibitively time consuming or expensive to fabricate using traditional manufacturing methods.

However, even with all of their advantages existing 3D printers still have drawbacks when it come to fabricating functional components. One of the main drawbacks is the limited selection of materials. Most widely used fused deposition modeling (FDM) 3D printers are limited to printing parts using thermoplastic materials such as ABS, PLA, or Nylon. While there are printers which are capable of printing certain metals and composite materials, the price and operating cost for these machine make them out of reach for most applications.

A Dimension SST 1200es FDM 3D printer was used for fabricating most of the 3D printed components used on the ATOM prototype. The printer is able to print components using *ABSplus* material and also has a second extrusion head for printing soluble support material. While *ABSplus* is a strong and durable plastic, it lacks the flexural rigidity needed for some of the components on the prototype such as the claw components that acted as the structural support for ball wheels. This led to deflections of the claw which could cause the platform to deviate from the desired motion. Design modifications to increase rigidity of the claws and minimize motion errors were implemented for the second generation prototype.

The first generation prototype utilized a triangular three ball-drive configuration rigidly mounted to a central chassis.

For the second generation prototype a diamond shaped four ball drive configuration was utilized and an articulated suspension system was added to each leg of the platform. The suspension system was designed and incorporated into the second generation prototype to increase the range of drivable terrains for ATOM.

After each mechanical modification, an effort to implement a measurement, calibration and motion control methodology was required. Prior research efforts to calibrate robots with additional sensors [8] in a closed-loop or open-loop form [9-10] exist. These methods use initial kinematic estimations and internal sensors already assembled or the addition of new sensors which are used only for testing. These methods are applied after robot is mechanically completed and used to test a control algorithm which was already developed. For the ATOM prototype a methodology was needed to measure dynamic system parameters of the robot empirically after each mechanical improvement before developing system for generating motion commands. In this paper, we share our experiences of modeling, calibrating and controlling the ATOM prototype using a systematic iterative empirical methodology.

Organization of the paper is as follows. Design modifications to increase rigidity of the claws and minimize motion errors are presented in section II. The next section discusses the design of the articulated suspension system for ATOM. Section IV lists an outline of the empirical methodology used during experiments, and the following two sections explain state space models and algorithms used to generate and execute motion plan. After presenting experimental setup and results, the paper is summarized in the conclusion section.

II. BALL SUPPORT STRUCTURE

The design of the structural support for the ball wheels has been one of the most challenging aspects of the mechanical design and development process for the ATOM prototype. In the first generation prototype described in [5] numerous design iterations were performed for the claw-like support structure of the ball wheels. The objective of the design was to securely hold the ball wheels in place without restricting any of their rotational degrees of freedom (DOF) while exposing as much of the surface area of the ball wheels to the ground terrain. After a number of design iterations the final design used for the first generation prototype consisted of 3D printed ABS plastic claw fingers with miniature omni-wheels used for contact with the ball wheels (Fig. 1). The miniature omni-wheels had a diameter of 1.5 inches and were also fabricated using a 3D printer since a commercially available omni-wheel of the desired size was not found. This design was able to successfully achieve many of the objectives specified for the first generation prototype.

During further testing of the first generation prototype, unwanted deflections of the claws were observed when the vehicle performed agile maneuvers which involved rapid accelerations and decelerations or sudden changes in direction. These deflections caused misalignments of the ball wheels resulting in deviations of the platform from the

desired motion. To alleviate this problem the structural rigidity of the claw fingers had to be improved.

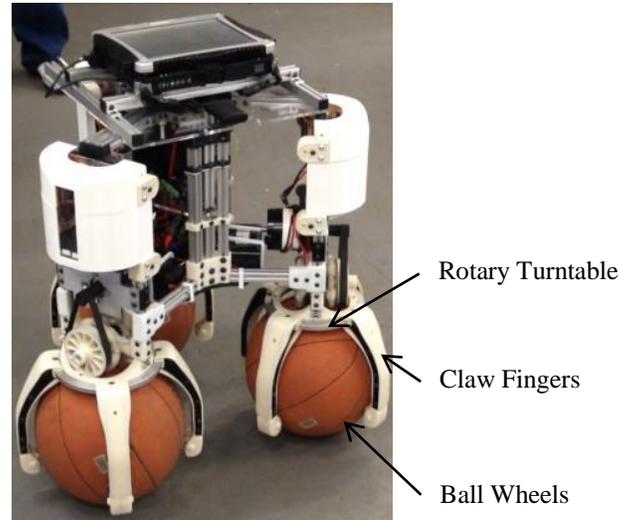


Fig. 1. First Generation Beta prototype of ATOM platform

It is possible to improve the rigidity of the claw by fabrication them using stiffer materials such as metals or composites, however, for the prototype it was decided to leverage the speed advantages of additive manufacturing instead. Therefore a brace was designed that could be incorporated into the existing claw-like support shown in Fig.2. The brace is composed of four parts that slide over each of the claw fingers and interconnect in the middle to form a rigid ring around the claws. A miniature omni-wheel is added on the brace in-between each claw finger to act as another point of contact between the claw support and the ball wheels. Small deflection of the claws are still observed during rapid acceleration and deceleration, however the brace has dramatically reduced the deflection of the claw fingers from before.

The claw fingers are mounted on a turntable which enables them to freely rotate around the ball wheels. If a claw finger comes in contact with an obstacle on the ground this turntable allows the claw fingers to rotate away exposing the surface of the ball to the obstacle. However, it was observed that this feature of the claw design may also contribute to motion errors. This occurs due to varying forces on the claw fingers, caused by the ball wheels, during motion which produce unwanted rotation of the claws. In order to eliminate this source of motion error, the turntable used for claw fingers was locked in place for the experiments performed. In future revisions of the claw a spring loaded mechanism will be used to hold the claws in place during motion while still enabling them to rotate out of the way during contact with an obstacle. This will minimize motion error from the turntable while maintaining the function of the turntable.

III. ARTICULATED SUSPENSION SYSTEM

Compared to omni-directional robot platforms that make use of omni-wheels or Mecanum wheels, the multi ball-drive system used on the ATOM prototypes is more capable of traversing over a wider range of terrains. This is mainly due

to the fact that small rollers on the omni-wheels are not in direct contact with the ground terrain and the isotropic geometry of the ball wheels enables them to roll in any direction with the same level of ease.

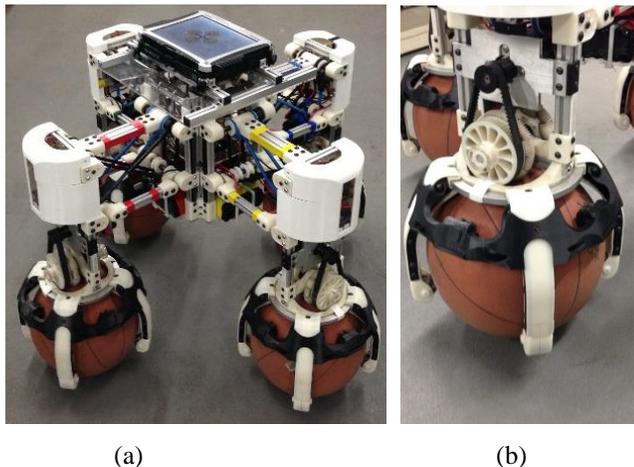


Fig. 2. (a) Second Generation Alpha prototype of ATOM Platform (b) close up of new claw design with additional brace for more rigidity

The ball drive units in the first generation prototype were rigidly mounted to the chassis without any suspension. This prototype was able to traverse over ground terrain and obstacles with variations up to 2 inches in height. Obstacles of greater height would cause the vehicle platform to tilt too much making it difficult for the omni-wheels to properly actuate the balls. A suspension system was designed and incorporated into the second generation prototype to increase the range of drivable terrain.

The suspension system that was chosen was a parallel four bar mechanism that was incorporated between the chassis and each ball drive unit. This articulated suspension system provides independent suspension with a large range of motion to each of ATOM’s legs (Fig. 3). The parallel four bar mechanism also insures that the ball-drive units are always properly aligned with the main chassis and enable the ability to change the height and footprint of ATOM.

The suspension for the four bar mechanism is provided by gas springs which are mounted diagonally from the bottom left hand joint of the mechanism to the top right hand joint. The gas springs support the weight of the platform and provide passive suspension during motion. Another reason that the four bar mechanism design was chosen was due to the fact that active suspension can also be incorporated into the mechanism. This can be accomplished by using a motor driven cable pulley system around the four joint of the mechanism in the shape of an infinity (∞) loop. By utilizing a variable compliance actuators in an antagonistic configuration [11] the motors for the cable pulley system would be able to control both the position as well as the stiffness of the legs. This can allow for advanced stability control of the vehicle by enabling adaptive suspension, where its stiffness can be adjusted for the terrain, as well as auto platform leveling where the legs can be articulated to level the platform when operating on uneven terrain.

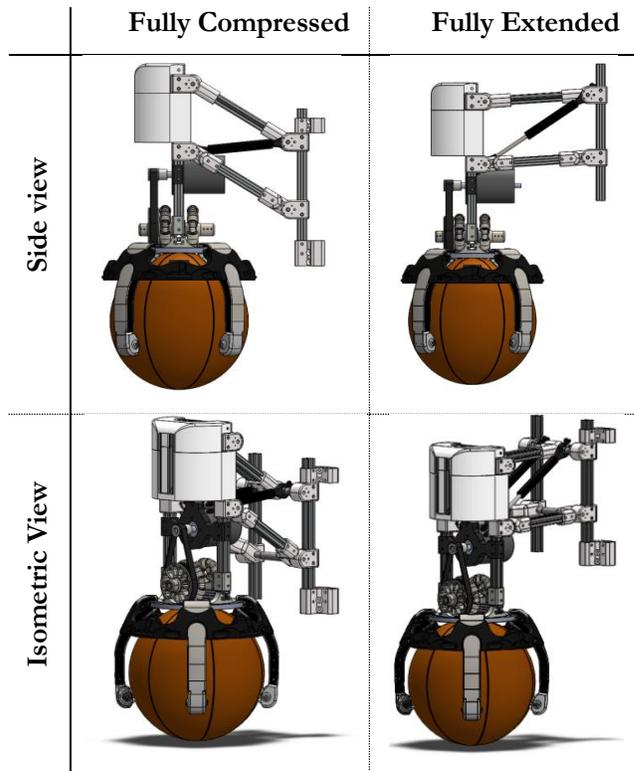


Fig. 3. The articulated parallel four bar suspension system used for second generation ATOM prototype

IV. SYSTEM CALIBRATION METHODOLOGY

Goal of this methodology is to probe and log robot’s trajectory under various motion profiles, systematically analyze and predict unknown kinematic components to tune up controller input for motion plan execution. An overview of the methodology is given in Table 1.

Initial values are crucial for experiments. Motion of the robot is dependent on torque generated by motors, ground frictions, inertias as well as supplied velocity profile, digital and analog circuit components used by the controller. All of these parameters combined define a single resultant force vector. To simplify this complex system it is modeled using a single acceleration parameter for each DOF.

TABLE I. ITERATIVE EMPIRICAL FINE TUNING STEPS

Steps	Iterative Empirical Fine Tuning Steps
1	Derive governing equations and define state space representation including controller input
2	Generate waypoints for planned path
3	Enter initial kinematic parameters and generate controller input
4	Apply motion plan and log data
5	Calculate tracking error
6	If residual is greater than preset value, go to step 3 and update initial kinematic parameters

The resultant force function might have second order components since parameters vary on the condition such as different moments generated. Also, jerk is observed in the motion. Even these force values will be updated later, limiting maximum values such as current supplied might reduce overall jerk occurred and save time to reach the optimized values.

After controller input is generated, motion plan is executed. During the experiment, an overhead camera system is used to measure position of robot to calculate maximum velocity, acceleration and deceleration values. This information is used to calculate residuals and adjust kinematics values.

State space representation of the robot, motion plan generation and residual calculations are discussed in detail in the following section.

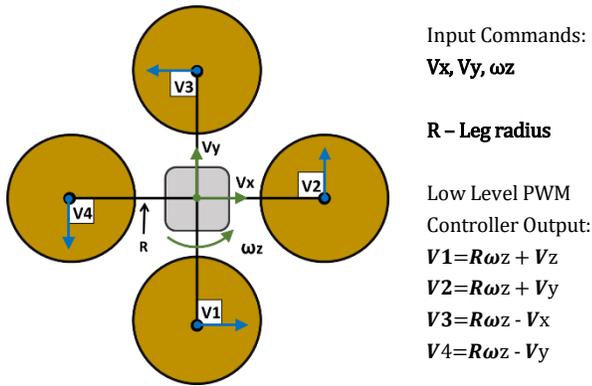


Fig. 4. Diagram and low level controller output for four ball configuration with diamond orientation

V. OMNIBALL SYSTEM MODEL

A. System State Vector

The robot's state includes position and heading as well as linear and angular velocity values as in

$$\mathbf{X}^R = [x^W \ y^W \ \theta^L \ V^{Wx} \ V^{Wy} \ \omega^L]^T \quad (1)$$

where superscript {W} represents world coordinate and {L} represents local coordinate while V and ω are linear and angular velocity respectively. Local coordinate is fixed on the robot's body and variation in heading is propagated to this frame. Initially both frames are coinciding.

B. State Transition

Control system uses four balls diamond orientation with motion vectors shown in Fig. 4. Forward motion is generated by side motors and lateral motion is controlled by front and rear motors. Heading change is generated by all motors. The state update equation is defined as in Eq. 2

$$F(\mathbf{X}_i^R, u_i) = \begin{bmatrix} x_i^W + \cos(\theta_i^L) (V_i^{Wx} \Delta t + 0.5 a_x \Delta t^2) \\ y_i^W + \sin(\theta_i^L) (V_i^{Wx} \Delta t + 0.5 a_x \Delta t^2) \\ R_i^L + \omega_i^L \Delta t + 0.5 a_R \Delta t^2 \\ V_i^{Wx} + a_x \Delta t \\ V_i^{Wy} + a_y \Delta t \\ \omega_i^L + a_R \Delta t \end{bmatrix} \quad (2)$$

where subscript {i} defines current state. Forward and angular acceleration components are generated by forces applied and values are observed with experiments.

C. Generating Motion Plan

First waypoints are defined in the system. Each waypoint has position and heading information. These points are used to generate intermediate plan for each control time stamp including linear and angular velocity values. Finally the motion commands applied to the embedded controller are represented by pulse-width modulation (PWM) values which are applied directly to the motors driving each ball. Flow of the calculations is given as in Eq. 3.

$$WP(x, y, \theta)_{1..k} \rightarrow P(Vx, Vy, \omega, \Delta t)_{1..n} \rightarrow PWM_{1..n}^{V1..V4} \quad (3)$$

where WP represents k^{th} input way points and P represents generated discrete points of planned trajectory with equal time { Δt } spacing. Finally these points are used to calculate low-level controller values.

D. Determining Motion Profile

There are two cases evaluated to generate controller input. First case is executed if algorithm detects that waypoints are far enough that robot can reach its maximum linear or angular velocity. Second case, which mostly occurs in rapid maneuvers, is executed if maximum speed can't be reached. The objective is to avoid any overshoot and minimize error while following the path.

Let's define t_1, t_2 and t_3 which represent acceleration, maximum velocity and deceleration durations as shown in Fig. 5 (case-I) and Fig. 6 (case-II). $\{a^+\}$ and $\{a^-\}$ represent acceleration and deceleration values, $\{V_0\}$ represents initial velocity, and $\{V\}$ in Fig.6 represents the instantaneous velocity reached during rapid motion. Angular motion profile plots have the same behavior.

The motion profile for each waypoint is generated using t_1, t_2 , and t_3 as shown in Fig. 5 and Fig. 6. Displacements for all DOF are modeled as in

$$|WP_{k,k-1}| = \int_0^{t_1} V^+(t) dt + \int_{t_1}^{t_2} V_{max} dt + \int_{t_2}^{t_3} V^-(t) dt \quad (4)$$

where $\{|WP|\}$ is total displacement required while traversing from waypoint {k-1} to {k}, $\{V\}$ is velocity profile and superscripts {+,-} represent acceleration and deceleration respectively. Same calculations are performed for all DOFs.

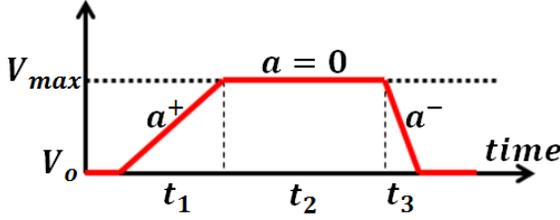


Fig. 5. Case-I motion profile. Velocity reaches to maximum and t_2 is greater than zero.

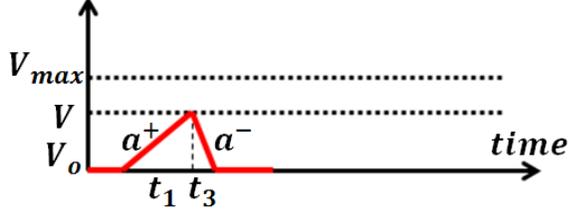


Fig. 6. Case-II motion profile. Velocity doesn't reach to maximum and t_2 doesn't exist. This profile is required for rapid motion.

When rapid maneuver is required, t_2 might not exist. First t_1 and t_3 are calculated based on the assumption that velocity will reach the maximum value for both linear and angular motion. If total displacement during t_1 and t_3 durations is smaller than total distance to the way point, it is guaranteed t_2 exists. Then, case-I is applied. Otherwise, case-II is selected. It is trivial to calculate t_1 and t_3 using measured acceleration and deceleration values.

E. Residuals and Waypoint Radius

At each step of control input generation algorithm, linear and angular residuals are calculated as in

$$E_{xy_i} = \sqrt{(x_i^W - WP_k(x))^2 + (y_i^W - WP_k(y))^2} \quad (5)$$

$$E_{\theta_i} = \theta_i^L - WP_k(\theta) \quad (6)$$

where WP_k represents current target waypoint. If the robot executes the required course change just at the moment it reaches to next waypoint, it will cause an overshoot or will cost additional time to adjust heading.

However, to generate a smooth motion and minimize time required for course adjustment, both errors could be minimized while robot is still traversing. Since t_3 is known, a waypoint radius is calculated as in

$$WR_k = |0.5 a^- t_3^2| \quad (7)$$

for both linear and angular values.

F. Generating Low-Level Control Signal

Motion plan algorithm iterates the solution till it reaches the final way point. At each step it calculates system linear and angular velocity values V^{Wx} , V^{Wy} , ω^L . These are then converted into PWM value as shown in Fig. 4 and applied to the drive motors.

G. Summary of Motion Plan Algorithm

Table.2 summarizes previous steps and defines the motion plan algorithm. It starts with reading waypoints and initializes the system with null values. Motion profiles to be applied between two waypoints are determined.

Then algorithm calculates error at each time step and updates the state by applying controller input. Updated state is used to generate low level controller input.

TABLE II. MOTION PLAN ALGORITHM

Steps	Motion Plan Algorithm
1	Read waypoints and initialize system state
2	For each way point, determine motion profile
3	Iteratively generate motion plan until final point is reached
3.1	Calculate linear and angular residuals
3.2	Generate controller input
3.3	Update state with generated linear and angular velocities
3.4	Generate low-level controller input
3.5	If waypoint radius is reached, get new waypoint

VI. EXPERIMENTAL SETUP

An overhead camera is setup to visually estimate pose of robot. Fig.9 illustrates a moving robot with superimposed previous positions. Ground markers (white) and pattern (yellow) are used for template tracking and matching to generate displacement and affine vector representing heading.

After initial observations, maximum linear velocities (V_x and V_y) are limited to 75% to minimize the jerk and maximum rotational velocity (ω_z) is limited to 25% to remain within acceptable speeds. Linear and angular maximum velocities are measured to be 1.143 m/s and 122.7°/s. While acceleration values are measured to be 0.65 m/s² and 43.82°/s², deceleration values are measured to be 0.95 m/s² and 92.25°/s².

Three types of experiments were performed to test the system parameters measured for the robot: A fixed point 90° rotation, 3 meters linear displacement and L-shaped 3x3 meter path with 90° change in heading at the corner. Motion plans were generated as shown in Fig. 7 and Fig. 8 which represent displacement, velocity and acceleration plots for both linear and angular motion. First case could be observed in Fig. 7 $\{V_x\}$ plot in blue which represents the three meters linear motion experiment. Velocity reaches maximum speed that acceleration goes to zero for a period of time. Also, in Fig. 8 $\{\omega\}$ plot which represents L shaped maneuver experiment shows that angular velocity doesn't reach its maximum value and a case-II motion profile is executed to track planned path.

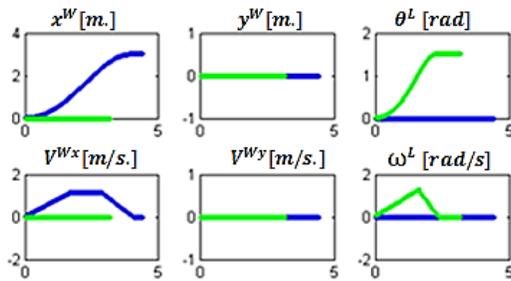


Fig. 7. Motion Profile for linear (blue) and angular (green) displacement experiments.

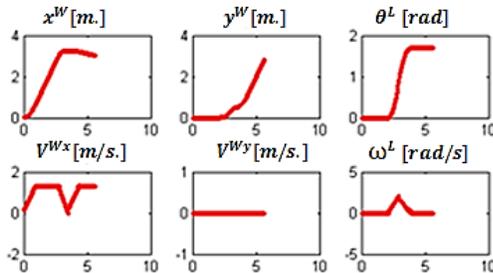


Fig. 8. Motion Profile for L shape path experiments.

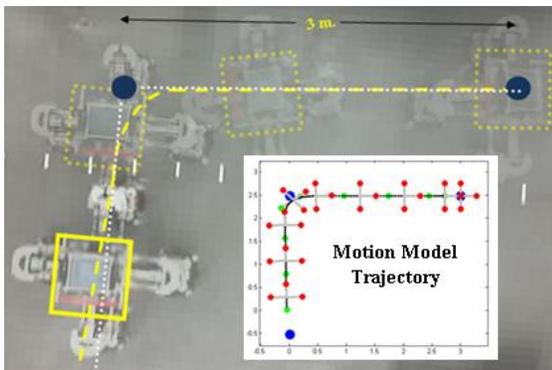


Fig. 9. Overhead camera view of experimental area with ground markers. Composed image illustrates multiple positions of robot. Blue circles: waypoints, white dotted line: planned path, yellow line: executed path.

VII. EXPERIMENTAL RESULTS

Each experiment was executed five times. Average total angular position error of $4^\circ (+/- 1.2^\circ)$ was measured for the 90° zero point rotation experiments and an average total displacement error of 12 cm. ($+/- 3$ cm.) was measured for the 3 meter linear displacement test. Fig.9 shows a result of maneuvering experiment with composed positions and superimposed followed trajectory. While total rotation error after turn is around $4.5^\circ (+/- 1.5^\circ)$, average position error on y axis after turn was around 15 cm ($+/- 6$ cm.).

VIII. CONCLUSIONS

The development of a second generation prototype for the ATOM ground robot platform was presented in this paper. Mechanical design modification to the support structure of the balls, which helped to improve the motion accuracy of the robot, as well as the addition of a new suspension system to help increase the range of drivable terrains were described.

The design complexity and high level of uncertainty present in the numerous system components make the accurate estimation of a robot's dynamic model challenging. Therefore, an empirical methodology to automatically generate motion commands for a given path was presented. This iterative methodology uses experimentally determined values for the whole system coupled with a simplified kinematic model to generate open loop motion commands for the robot. Test results have shown that this methodology is able to execute motion commands with a sufficient level of accuracy comparable to a skilled operator manually controlling the robots.

The performance of this methodology can further be improved by implementing closed loop controllers which can be used to minimize residual errors during motion. While an accurate dynamic model can be produced for a real system utilizing predictable system components, the developed empirical methodology has proven to be useful for generating motion commands for the prototype systems which utilize 3D printed components.

REFERENCES

- [1] Fiorini, P., "Ground mobility systems for planetary exploration," *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol.1, no., pp.908,913 vol.1, 2000 doi: 10.1109/ROBOT.2000.844164
- [2] Ferriere, L.; Raucant, B., "ROLLMOBS, a new universal wheel concept," *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol.3, no., pp.1877,1882 vol.3, 16-20 May 1998
- [3] Ghariblu, H., "Design and Modeling of a Ball Wheel Omni-directional Mobile Robot," *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, vol.2, no., pp.571,575, 19-21 March 2010
- [4] Wada, M.; Asada, H.H., "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs," *Robotics and Automation, IEEE Transactions on*, vol.15, no.6, pp.978,989, Dec 1999
- [5] Gebre, B.A.; Ryter, Z.; Humphreys, S.R.; Ginsberg, S.M.; Farrell, S.P.; Kauffman, A.; Capon, W.; Robbins, W.; Pochiraju, K., "A multi-ball drive for omni-directional mobility," *Technologies for Practical Robot Applications (TePRA), 2014 IEEE International Conference on*, vol., no., pp.1,6, 14-15 April 2014
- [6] Nagarajan, U.; Kantor, G.; Hollis, R.L., "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, vol., no., pp.3743,3748, 12-17 May 2009
- [7] Hertig, L.; Schindler, D.; Bloesch, M.; Remy, C.D.; Siegwart, R., "Unified state estimation for a ballbot," *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, vol., no., pp.2471,2476, 6-10 May 2013
- [8] Wampler, C.W.; Hollerbach, J.M.; Arai, T., "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms," *Robotics and Automation, IEEE Transactions on*, vol.11, no.5, pp.710,724, doi: 10.1109/70.466613, Oct 1995
- [9] Ciarleglio, C., "Kinematic Determination of an Unmodeled Serial Manipulator by Means of an IMU." PhD diss., 2013.
- [10] Mittendorf, P.; Cheng, G., "Open-loop self-calibration of articulated robots with artificial skins," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, vol., no., pp. 4539-4545, May 2012
- [11] Laffranchi, M.; Sumioka, H.; Sproewitz, A.; Gan, D.; Tsagaraki N.G., "Compliant Actuators," Adaptive Modular Architectures for Rich Motor Skills (AMARSi), March 2011