Trajectory Tracking Control for a Biomimetic Spherical Robot Based on ADRC

Mugen Zhou¹, Shuxiang Guo¹,²*, Xihuan Hou¹, He Yin¹,
Ao Li¹, Debin Xia¹, Zan Li¹, Meng Liu¹
¹Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, School of Automation, Beijing Institute of Technology
No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China
²Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa, Japan
zhoumugen@bit.edu.cn; guoshuxiang@bit.edu.cn
*Corresponding author

Abstract—In this article, a two-dimensional trajectory tracking control framework is proposed for biomimetic spherical robots (BSR) in a constrained workspace despite the lack of dynamic model parameters information and the effects of disturbances on the robot motion. Meanwhile, the research presents the general dynamics models of the robot and the thrusters allocator scheme to ensure the force generated by the propellers within the feasible range. Our trajectory tracking control framework relies on three active disturbance rejection controllers (ADRC) for the case of biomimetic spherical robots. More importantly, the controllers consider various practical operational constraints, such as continuous and smooth controller outputs, predefined velocity bounds, and thruster saturations. Finally, We assess the performance and feasibility of the proposed control framework through the simulations.

Index Terms—Biomimetic spherical robot, Trajectory tracking control, Active disturbance rejection control.

I. INTRODUCTION

In recent years, a growing number of scientists focus their attention on marine robots, including autonomous underwater vehicles (AUVs) [1], [2], unmanned surface vehicles (USVs) [3], [4], remotely operated vehicles (ROVs) [5], and biologically inspired robots [6]–[8] for commercial, military, and academic research applications [9]. These robots provide a wide range of capabilities such as underwater inspection and surveillance, ecosystem monitoring, underwater exploration [10]. Many of these applications require the marine robotic vehicles to operate accurately in a constrained workspace with disturbances of the environment and uncertainties of the model.

In particular, Accurate trajectory tracking control is essential for underwater robot applications, but designing a robust trajectory tracking is a challenging work due to uncertain model parameters, unknown disturbances, and various constraints [11]. As a result, there has been plenty of studies on this challenging topic in the past decade. The trajectory tracking problem is defined by controlling the vehicle reach and follow a desired trajectory, which is coupled with time assignments [9]. In comparison with path-following, the desired trajectory of trajectory tracking is dependent on time.

The most popular method for trajectory tracking control is sliding mode control (SMC), unlike conventional methods that are sensitive to parameter changes or rely strongly on accurate model parameters. Jiang et al. develop an adaptive robust integral sliding model controller to avoid the jump problem of the velocity by using bioinspired neurodynamics [12]. Zhou et al. integrate the backstepping method and radial basis function (RBF) neural network in the sliding mode controller to estimate the uncertainty of model parameters and the effect of external disturbances [13]. To provide the robustness and adaptation of the controller, Yan et al. design dual closed-loop integral sliding mode controllers, including a velocity loop and a position loop [14]. Without any prior knowledge of uncertainty and disturbance, sliding mode control method coupled with adaptive nonsingular integral terminal is proposed by Qiao et al. to improve the convergence speed [15]. Due to the input delay, an integrated time-delay sliding mode control strategy is designed in [16]. However, the ”chattering” problem still exists, which may yield high-frequency dynamics [17].

Model predictive control (MPC) for trajectory tracking is an alternative method that combines the model dynamics and allows for minimal adjustment of controller by minimizing object function [3]. Meanwhile, MPC is an ideal tool to handle internal constraints, such as thrust saturation, velocity increment, and acceleration constraints and external constraints, including safe operating area and external disturbance [18], [19]. Hu et al. design a trajectory tracking controller using MPC to avoid obstacles [18]. Due to actuator saturation, a Lyapunov-based model predictive control (LMPC) framework is proposed by Chao et al. to improve the tracking performance and save computational resources [19]. An improved model predictive control method is proposed by Hou et al. to process practical constraints and thrusters saturation [20]. Chao et al. develop a distributed NMPC algorithm to reduce the floating point operations [11]. A nonlinear model predictive dynamic positioning strategy to account...
for complex stochastic disturbances in [5]. Nevertheless, the hydrodynamic parameters identification for underwater robots is not easy to perform and the computational resources and time consumption of the MPC algorithm are high [3].

The observer has been widely applied to estimate internal model uncertainty and external environmental disturbance in robotics research field. With the help of the extended state observer (ESO), Peng et al. recover unmeasurable velocities and estimate total disturbance caused by internal uncertainty and external interference [21]. Qin and estimate total disturbance caused by internal uncertainty and external interference [21]. Qin et al. design a disturbance observer to deal with the effects of the external disturbances, especially thruster faults [22]. Lakhekar et al. combine the disturbance-observer-based control with fuzzy adapted S-Surface control to compensate the unknown disturbances and unmodeled dynamics [23]. Wang et al. propose a control architecture, which includes three active disturbance rejection control (ADRC) sub-controllers to allow for 3-D helical path following of an underwater biomimetic vehicle [24]. Inspired by the above motion control schemes, in particular, by ADRC [24], this paper aims to develop a trajectory tracking control scheme by using three ADRC controllers with internal uncertainties and external disturbances in a constrained workspace.

The rest of this article is organized as follows. The next section introduces the modeling of our biomimetic spherical robot [6], [25], [26] and the thrust allocation scheme. In Section III, the trajectory tracking control framework is detailed. To evaluate the robustness of the controller, we perform simulations for straight and circular trajectories in Section IV. Finally, the conclusion is provided in Section V.

II. MODELING OF THE BIOMIMETIC SPHERICAL ROBOT

A. Modeling of the Biomimetic Spherical Robot

![Fig. 1. The world reference frame and the body-fixed reference frame.](image)

To analyze the trajectory tracking clearly, the inertial reference frame \((O_E - X_E Y_E)\) and the body-fixed reference frame \((O_B - X_B Y_B)\) are described in Fig.1. The desired trajectory is described as \([x_r(t), y_r(t), \psi_r(t)]^T\). In the world frame, states of BSR are expressed as:

\[
\eta = [x, y, \psi]^T
\]  

where \((x, y)\) are the plane coordinates and the \(\psi\) is the yaw angle correspond to the surge orientation with respect to the positive x axis. In the body-fixed frame, motion states of the robot are expressed as:

\[
v = [u, v, r]^T
\]

where, \(u, v, r\) are the linear velocity in the surge, sway, and the yaw rate, respectively. And \(\beta = \arctan(e_y/e_x)\) is the azimuth angles, where \(e_x = x_r - x, \ e_y = y_r - y\) are the tracking error in the x-direction and y-direction, respectively.

In this paper, the trajectory tracking for biomimetic spherical robot, focuses on a two-dimensional horizontal control. The kinematic model of the BSR is described as follows:

\[
\begin{align*}
    \dot{v} &= J(\psi)\dot{\psi} \\
    \dot{\psi} &= J(\psi)(\hat{\psi} + \hat{\psi}\dot{\psi})
\end{align*}
\]  

where \(\hat{\psi} = [\dot{x}, \dot{y}, \dot{\psi}]^T\) denotes the velocities of the BSR in x-direction, y-direction and the angular velocity around z-direction in the inertial frame, respectively; \(\hat{\psi} = [\dot{x}, \dot{y}, \dot{\psi}]^T\) represents the corresponding accelerations and angular acceleration in the inertial frame, respectively; \(\dot{\psi} = [u, v, r]^T\) refers to the corresponding accelerations and angular acceleration respectively. \(J(\psi)\) is the rotation matrix depending on the yaw angle \(\psi\):

\[
J(\psi) = \begin{bmatrix}
    \cos(\psi) & \sin(\psi) & 0 \\
    -\sin(\psi) & \cos(\psi) & 0 \\
    0 & 0 & 1
\end{bmatrix}
\]

\[
\hat{\psi} = \begin{bmatrix}
    0 & \psi & 0 \\
    -\psi & 0 & 0 \\
    0 & 0 & 0
\end{bmatrix}
\]

As the low speed of the BSR motion, the nonlinear damping is neglected since the linear damping outweigh the nonlinear damping greatly. Considering the distribution of thruster, the BSR with \(H\) configuration is unable to generate force in the sway direction in that the control of the BSR is underactuated. The dynamic model is expressed as equation (4):

\[
M \ddot{v} + C(v)v + Dv = \tau + \tau_d
\]

With

\[
\tau = [\tau_u, 0, \tau_r]^T \quad \tau_d = [\tau_{du}, \tau_{dv}, \tau_{dr}]^T
\]

\[
M = \begin{bmatrix}
    m_{11} & 0 & 0 \\
    0 & m_{22} & 0 \\
    0 & 0 & m_{33}
\end{bmatrix} \quad D = \begin{bmatrix}
    d_{11} & 0 & 0 \\
    0 & d_{22} & 0 \\
    0 & 0 & d_{33}
\end{bmatrix}
\]

\[
C(v) = \begin{bmatrix}
    0 & 0 & -m_{22}v \\
    0 & 0 & m_{11}u \\
    m_{22}v & -m_{11}u & 0
\end{bmatrix}
\]

where \(M\) is the inertial matrix including hydrodynamic additional mass. \(C(v)\) is the coriolis and centripetal matrix.
\(D\) is the linear hydrodynamic damping matrix. \(\tau\) is the input vectors, where \(\tau_u, \tau_r\) describe the propulsive force and force moment, respectively. \(\tau_d \in \mathbb{R}^3\) represents the nonlinear external disturbances on surge, sway and yaw, which are supposed to be time-varying bounded.

### B. Thrust Allocation Scheme

Considering the response and saturation characteristics of the propellers, a reasonable thrust allocation scheme is designed to make the BSR motion more stable.

![Fig. 2. The propeller distribution of the BSR with H configuration.](image)

The applied force \(\tau_u\) and force moment \(\tau_r\) are defined as equation (5):

\[
\begin{align*}
\tau_u &= -f_1 + f_2 + f_3 - f_4 \\
\tau_r &= (-f_1 + f_2 - f_3 + f_4) \cdot l \sin \alpha
\end{align*}
\]  

(5)

where \(f_1, f_2, f_3\) and \(f_4\) are the forces generated by the corresponding propellers, as shown in Fig.2(a). Each propeller is fixed and generates forces opposite to the direction of the water outlet. \(2a = 0.055(m)\) is the distance between the transverse propellers, \(\alpha\) is the angle between the line of the longitudinal thrusters and of the diagonal thrusters, \(l\) is the distance between the propeller and the geometric center, as shown in Fig.2(b).

Further, the input forces \(F = [f_1, f_2, f_3, f_4]^T\) for each propeller can be written as:

\[
F = \begin{pmatrix}
1 & -\frac{1}{4} & -\frac{1}{4a} \\
1 & \frac{1}{4} & \frac{1}{4a} \\
1 & \frac{1}{4} & -\frac{1}{4a} \\
1 & -\frac{1}{4} & \frac{1}{4a}
\end{pmatrix}
\begin{pmatrix}
f_{\text{static}} \\
\tau_u \\
\tau_r
\end{pmatrix}
\]  

(6)

where \(f_{\text{static}} = 1(N)\) is a initial force, which can avoid sudden propeller start and speed jump. When every propeller has an same angular speed \(\omega_{\text{static}}(\neq 0)\) and generates same force \(f_{\text{static}}\), the total force and force moment vector \([\tau_u, \tau_r]^T\) applied to robot so the robot remains still.

It is worth pointing out that the propellers take a certain response time to reach the desired speed, so the the propellers can not start suddenly and jump in speed. In addition, the propellers exit dead zone and saturation zone. Taking into consideration the aforementioned constraints, we define \(f_{\text{min}} = 0\) and \(f_{\text{max}} = 2f_{\text{static}}\) as the lower and upper bounds of \(f_i\), where \(i = 1, 2, 3, 4\), respectively. Thus, we obtain the following constraints on the input force matrix:

\[
\begin{align*}
f_{\text{min}} &\leq f_i \leq f_{\text{max}} \\
0 &\leq f_i \leq 2f_{\text{static}}
\end{align*}
\]  

(7)

So the input vectors \([\tau_u, \tau_r]^T\) meet the Manhattan Distance constraint as equation (8). When the desired force or moment exceeds the feasible range, the desired force and moment vector is scaled proportionally to make it satisfy the constraint.

\[
|\tau_u| + |\tau_r| \leq 4f_{\text{static}}
\]  

(8)

### III. Trajectory Tracking Control Methodology

In this section, we give an overview of the trajectory tracking framework, which is composed of three main modules: position controller (Section III-A), yaw controller (Section III-B), and the force allocator (Section II-C), as shown in Fig.3. The input of the trajectory tracking framework is a global reference trajectory predefined and the operation constraints include predefined velocity bounds and thruster saturations. The position controller and the yaw controller all based on the ADRC controller computes the corresponding force and force moment vector that satisfies the operation constraints, respectively. The force allocator based on dynamics can map the force and moment into the forces of each propeller. Finally, the global camera provides an estimation of the robot position and yaw angle.

![Fig. 3. Overview of the trajectory tracking control framework.](image)

**A. Position controller**

The position controller is composed of X position controller and Y position controller, which control \(x\) and \(y\) in the inertial reference frame.
Based on the model of the BSR, the dynamic equation of position in the X-direction can be represented as follows:

\[
\begin{align*}
\dot{x} &= v_x \\
\ddot{x} &= f_x + b_x \tau_x
\end{align*}
\]

(9)

with

\[
\begin{align*}
f_x &= -\ddot{y} + \frac{\cos(\psi)}{m_{11}} (m_{22} \ddot{v} - d_{11} u + \tau_{du}) \\
&\quad + \frac{\sin(\psi)}{m_{22}} (m_{11} \ddot{v} + d_{22} v - \tau_{dv}) \\
b_x &= \frac{1}{m_{11}} \tau_x = \cos(\psi) \tau_u
\end{align*}
\]

(10)

where \( f_x \) is a summation of the internal and external disturbance. \( b_x \) denotes the control gain. Specifically, Fig.4 shows that a tracking differentiator is used to generate the tracking signal \( x_d \) and the differential signal \( x_c \) of the x-axis coordinate value \( x_r \) of the target point on the trajectory. An ESO outputs the estimations (ie., \( \zeta_x1, \zeta_x2, \zeta_x3 \)) of the systems states \( x, \dot{x} \), and the general disturbance \( f_x(\cdot) + (b_x - b_x0) \tau_x \) based on the position value feedback. by actively compensating \( f_x \) using \( \zeta_x3 \), the control law for equation (11) is given by:

\[
\tau_x = \frac{\tau x0 - \zeta_x3}{b_x}
\]

(11)

Therefore, the equation (9) is simplified to a double integrator:

\[
\dot{x} \approx \tau x0
\]

(12)

A proportional-derivative controller is designed to control it:

\[
\tau x0 = k_xp e_{x1} + k_xd e_{x2}
\]

(13)

where \( k_xp \) and \( k_xd \) are the proportional gain and derivative gain, respectively. \( e_{x1} = x_1 - \zeta_x1 \) and \( e_{x2} = x_2 - \zeta_x2 \) are states errors. The differential signal subjects to the acceleration limit of \( \dot{\delta}_x \). More details about implement the ADRC are described in [24].

And the dynamic equation of position in the Y-direction can be represented as follows:

\[
\begin{align*}
\dot{y} &= v_y \\
\ddot{y} &= f_y + b_y \tau_y
\end{align*}
\]

(14)

with

\[
\begin{align*}
f_y &= \ddot{y} + \frac{\sin(\psi)}{m_{11}} (m_{22} \ddot{v} - d_{11} u + \tau_{du}) \\
&\quad + \frac{\cos(\psi)}{m_{22}} (-m_{11} \ddot{v} + d_{22} v - \tau_{dv}) \\
b_y &= \frac{1}{m_{11}} \tau_y = \sin(\psi) \tau_u
\end{align*}
\]

(15)

Since the ADRC controller to force BSR to track \( y_r \) is similar with tracking \( x_r \), the detailed process is not described again. Moreover, by combining the control signal in X and Y direction, the surge control is written as:

\[
\tau_u = \tau_x \cos(\psi) + \tau_y \sin(\psi)
\]

(16)

B. Yaw controller

Based on the model of the BSR, the dynamic equation of yaw motion can be described as follows:

\[
\begin{align*}
\dot{\psi} &= \omega \\
\dot{\psi} &= f_\psi + b_\psi \tau_\psi
\end{align*}
\]

(17)

with

\[
\begin{align*}
f_\psi &= -\frac{d_{33}}{m_{33}} \dot{\psi} + \frac{m_{11} - m_{22}}{m_{33}} \omega \psi_0 + \frac{1}{m_{33}} \tau_{dr} \\
b_\psi &= \frac{1}{m_{33}} \tau_\psi = \tau_r
\end{align*}
\]

(18)

where \( f_\psi \) represents the general disturbance including the external disturbance and internal dynamics. It is worth noting that the desired signal of the tracking differentiator is the azimuth angles \( \beta \) rather than \( \psi_r \), as shown in Fig.5. Yaw controller is analogous to the other two controller and therefore the detailed expressions are omitted here.

IV. SIMULATION AND RESULTS

The section presents two simulations of the BSR, using the developed method of trajectory tracking. The goal of the simulations is to illustrate the performance and validate the effectiveness of the controller. The dynamic model parameters and the disturbance forces applied to the BSR in the simulations are shown in Table I. The disturbance forces on
both surge and sway are modeled as uniform random noise with zero mean and variance of 0.1. The disturbance forces on yaw are defined as uniform random noise with zero mean and variance of 0.008. The update frequency of the controller is set to 50Hz. The parameters of position and yaw controllers are listed in Table II.

A. Straight trajectory tracking

The straight trajectory is generated with a surge velocity of $v_r = 0.15$ m/s and initial yaw angle of $\theta_r = \frac{\pi}{4}$. The start point is (-2,-2) and the robot starts at point (-1,-2).

Fig. 6. Performance of the straight trajectory tracking.

The straight trajectory is generated with a surge velocity of $v_r = 0.15$ m/s and initial yaw angle of $\theta_r = \frac{\pi}{4}$. The start point is (-2,-2) and the robot starts at point (-1,-2).

Fig. 7. The errors of BSR while tracking straight trajectory.

Fig.6 depicts the time evolution of the position for the BSR tracking the desired trajectory. The red line denotes the actual position and the blue line the desired position. Notice that the controller tracks the straight trajectory perfectly in simulation despite the unknown disturbance. The tracking error of the BSR while tracking the straight trajectory are shown in Fig.7. $e_d = \sqrt{e_x^2 + e_y^2}$ converges to 0.087 m and the maximum $e_\theta$ is just 0.0016 rad. More importantly, the surge speed of the BSR converges to 0.15 m/s, which is equal to $v_r$.

B. Circular trajectory tracking

The angular velocity $\omega_r$ of the circular trajectory is 0.1rad/s and radius $R$ 1m. The center and the start point of the trajectory are (0,0) and (1,0), respectively. The robot starts at the origin (0,0) with initial velocity $v_0(0m/s)$ and angle velocity $\omega_0(0m/s)$.

Fig.8 depicts the BSR tends to the reference trajectory and reaching the reference point, finally. The red line denotes the actual trajectory and the blue line represents the reference trajectory. Fig.9 shows $e_d$ converges to 0.059 m and $e_\theta$ tends to 0.030 rad. The velocity in surge and yaw angle converges to the speed of the reference trajectory. But because of the noise, the yaw angle rate oscillates at low frequency.

V. Conclusions

This article presents the kinematics and dynamics modeling of the biomimetic spherical robot, and the design of the trajectory tracking control framework. The thrust allocation scheme addresses the problem that the propellers can not start suddenly and jump in speed and takes into consideration the dead zone and saturation zone. More importantly, the trajectory tracking framework integrating ADRC is designed to reduce the influence of unknown disturbances and eliminates the dependence on accurate robot model parameters.
The simulation results suggest that the ADRC controller achieve accurate trajectory tracking and ensure smoothness and continuity of controller outputs.

ACKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China (61773064, 61503028), National Key Research and Development Program of China (2017YFB1304404), and National High Tech Research and Development Program (863 Program) of China (No.2015AA043202).

REFERENCES


