A Report on Enhanced Bioinspired Backstepping Control for a Mobile Robot With Unscented Kalman Filter

Hamza Shah Khan, ***UID: 119483152*****, Navdeep Singh,** ***UID: 120098024***

University of Maryland, College Park

Date: 11/14/2023

Abstract: This report describes our implementation of the work done by Yang et al.[1] on the development of an advanced control strategy that combines torque controllers and backstepping inspired by biology, together with state estimation using Kalman Filter (KF) and Unscented Kalman Filter (UKF). The torque and backstepping controllers that have been developed mitigate the typical issues that arise with traditional controls, guaranteeing smooth velocity commands and reducing overshoots.

Our main goal was to accurately estimate the robot's state by monitoring its linear and angular velocities through the use of a custom Kalman Filter. The robot's wheel torques were adjusted for output velocity matching by applying torque controllers after the obtained velocities were utilized as feedback to calculate differences from reference values.

Index Terms: Unscented Kalman Filter, Kalman Filter, Backstepping Controller.

1. Introduction

In robotics, real-time tracking control has been a key field of study, necessary to allow robots to navigate challenging environments and achieve accurate trajectory tracking. The difficulties are not only in ensuring smooth velocity changes but also in minimizing the effects of measurement and system noise on tracking accuracy. The paper "Enhanced Bioinspired Backstepping Control for a Mobile Robot with Unscented Kalman Filter" presents a novel approach that combines a torque controller with Kalman Filter (KF) and Unscented Kalman Filter (UKF) and a biologically inspired backstepping controller in an effort to improve tracking control strategies.

The study highlights the usefulness of Yang et al.'s bioinspired backstepping controller in addressing velocity jumps among other tracking control techniques. Incorporating UKF, which is specifically tailored for nonlinear systems, guarantees precise state estimations even when noise and dynamic limitations are present. This research provides a useful and efficient tracking control approach, which is especially beneficial for autonomous mobile robots operating in challenging environments.

This report documents the derivation of state equations, the development of torque and backstepping controllers, and the integration of UKF in order to add to the existing body of knowledge as an application of this innovative research. An in-depth discussion of the background, the suggested bio-inspired backstepping and torque control approach, simulation results, and a final summary of the findings are provided in the following sections. The objective of this implementation is to verify the efficiency of the suggested control technique and demonstrate how well it works in real-world situations by reducing velocity jumps, improving trajectory tracking, and improving resistance to external noise.

2. Background

2.1. Kalman Filter(KF) and Unscented Kalman Filter(UKF) Algorithms

A linear system with noises is given by:

$$x_{k+1} = Ax_k + Buk + \alpha_k \tag{1}$$

$$z_{k+1} = Hx_{k+1} + \beta_{k+1} \tag{2}$$

where A is the system matrix, B is the input matrix, u_k is the input, and H is the measurement matrix. Both system and measurement noises are treated as Gaussian, where $P(\alpha_k) \sim N(0, Q_k)$ and $P(\beta_k) \sim N(0, R_k)$. The processes of the Kalman Filter (KF) are defined as follows:

$$\hat{x}_{k+1|k} = Ax_{k|k} + Buk \tag{3}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \tag{4}$$

where $\hat{x}_{k+1|k}$ and $x_{k|k}$ are the priori and posterior state estimates, and $P_{k+1|k}$ is the corresponding state error matrix, which is the covariance of the process noise. Q_k is the system noise covariance. The updating stage is defined as:

$$K_{k+1} = P_{k+1|k}H^T \left[HP_{k+1|k}H^T + R_{k+1}\right]^{-1} \tag{5}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left[z_{k+1} - H\hat{x}_{k+1|k}\right] \tag{6}$$

$$P_{k+1|k+1} = \left[I - K_{k+1}H\right]P_{k+1|k} \tag{7}$$

Here, Kalman gain K_{k+1} is used to minimize the trace of the posteriori state error covariance matrix. $\hat{x}_{k+1|k+1}$ is the updated state estimate, $P_{k+1|k+1}$ is the posteriori state error covariance, and R_{k+1} is the measurement noise covariance.

In real-world applications, where linear systems are not perfect, the Unscented Kalman Filter (UKF) is used to handle nonlinear systems. For a general nonlinear system:

$$x_{k+1} = f(x_k, u_k) + \delta_k \tag{8}$$

$$z_{k+1} = h(x_{k+1}) + \gamma_{k+1} \tag{9}$$

In the context of the UKF, $f(x_k, u_k)$ represents the nonlinear system process, and h is the measurement model. The terms δ_k and γ_k denote Gaussian process and measurement noises, respectively. The initial step of the UKF involves creating $2n + 1$ sampling points, referred to as sigma points, where n signifies the dimension of the state variable x_k.

The state estimation for the mobile robot is computed using sigma points through a combination of weighting and multiple nonlinear functions. The initial sigma point, denoted as $x_{0,k|k}$, and its weight, W_0, are determined by:

$$x_{0,k|k} = \hat{x}_{k|k} \tag{10}$$

$$W_0 = \frac{\lambda}{n+\lambda} \tag{11}$$

Here, λ is a design parameter controlling the spread of sigma points, typically set significantly less than 1. The subsequent points, from the second to the $(n + 1)$-th point, are defined as:

$$X_{i,k|k} = \hat{x}_{k|k} + \sqrt{(n+\lambda)P_{k|k}} \cdot i \tag{12}$$

$$W_i = \frac{1}{2(n+\lambda)} \tag{13}$$

where $X_{i,k|k}$ and W_i are the corresponding sigma point and weight for the i-th sigma point. The final n sigma points are given by:

$$X_{i+n,k|k} = \hat{x}_{k|k} - \sqrt{(n+\lambda)P_{k|k}} \cdot i \tag{14}$$

$$W_{i+n} = \frac{1}{2(n+\lambda)} \tag{15}$$

From equations (10) to (15), the predicted state estimate, $\hat{x}_{i,k+1|k}$, is calculated as:

$$\hat{x}_{i,k+1|k} = f(X_{i,k|k}, u_k) \tag{16}$$

$$\hat{x}_{i,k+1|k} = \sum_{i=0}^{2n} W_i \hat{x}_{i,k+1|k} \tag{17}$$

The predicted state error covariance, $P_{k+1|k}$, is obtained through:

$$P_{k+1|k} = \sum_{i=0}^{2n} W_i (\hat{x}_{i,k+1|k} - \hat{x}_{k+1|k})(\hat{x}_{i,k+1|k} - \hat{x}_{k+1|k})^T + Q_k \tag{18}$$

All generated samples are then utilized to predict measurements using a nonlinear measurement model, defined as:

$$Z_{i,k+1|k} = h(X_{i,k+1|k}, u_k) \tag{19}$$

$$\hat{z}_{k+1|k} = \sum_{i=0}^{2n} W_i Z_{i,k+1|k} \tag{20}$$

The variables $\hat{z}_{i,k+1|k}$ and $\hat{z}_{k+1|k}$ represent the i-th measurement and the predicted measurement, respectively. Subsequently, these variables are used to calculate the measurement covariance, $P_{zz,k+1|k}$, and cross-covariance, $P_{xz,k+1|k}$, defined as:

$$P_{zz,k+1|k} = \sum_{i=0}^{2n} W_i (Z_{i,k+1|k} - \hat{z}_{k+1|k})(Z_{i,k+1|k} - \hat{z}_{k+1|k})^T + R_k \tag{21}$$

$$P_{xz,k+1|k} = \sum_{i=0}^{2n} W_i (\hat{x}_{i,k+1|k} - \hat{x}_{k+1|k})(Z_{i,k+1|k} - \hat{z}_{k+1|k})^T \tag{22}$$

Finally, the Kalman gain, state estimate, and state error covariance are obtained through:

$$K_{k+1} = P_{xz,k+1|k} P_{zz,k+1|k}^{-1} \tag{23}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - \hat{z}_{k+1|k}) \quad (33) P_{k+1|k+1} = P_{k+1|k} - K_{k+1} P_{zz,k+1|k} K_{k+1}^T \tag{24}$$

The UKF provides accurate state estimates for nonlinear systems without the need for linearization, making it preferable to Extended Kalman Filter (EKF).

2.2. Kinematics and Dynamics Model of Mobile Robot

The orientation of the non-holonomic robot in the inertial frame is represented by $P_c = [x_c, y_c, \theta_c]$, where x_c and y_c represent the spatial position at point C of the mobile robot, and θ_c is the orientation angle with respect to point C. The non-holonomic kinematic constraint imposed on the mobile robot is expressed as:

$$\dot{y}_c \cos\theta_c - \dot{x}_c \sin\theta_c = 0 \tag{25}$$

This constraint dictates that the robot moves perpendicular to the driving wheels. From a kinematic control perspective, disregarding slipping conditions, the mobile robot exhibits 2 degrees of freedom (DOF). The correlation between the velocity of the mobile robot in the inertial frame $\dot{P}_c = [\dot{x}_c, \dot{y}_c, \dot{\theta}_c]$ and the velocity in the body-fixed frame is established through a Jacobian matrix given by:

$$\dot{P}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_c \\ \omega_c \end{bmatrix} \tag{26}$$

where ν_c and ω_c represent the linear and angular velocity of the mobile robot.

The dynamic equation for the non-holonomic mobile robot, neglecting the influence of gravity, can be expressed as follows:

$$\bar{M}(\dot{P}_c)\dot{V} + \bar{F}(\dot{P}_c, \dot{P}_c)V + \bar{\tau}_d = \bar{B}\tau \tag{27}$$

where $\bar{M}(\dot{P}_c)$ is the positive definite matrix for inertia, $\bar{F}(\dot{P}_c, \dot{P}_c)$ is the centripetal and Coriolis matrix, $\bar{\tau}_d$ is the disturbance, and $V = [\nu_c, \omega_c]^T$.

Considering τ_d as an external disturbance, the simplified dynamics of the mobile robot are expressed as:

$$\dot{V} = \bar{M}^{-1}\bar{B} \cdot \tau_c \tag{28}$$

The mobile robot's dynamics are modeled by matrices:

$$\bar{M} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \bar{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \bar{B} = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ l & l \end{bmatrix}, \tau_c = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \tag{29}$$

where m is the mass of the mobile robot, I is the inertia, τ_R and τ_L are the right and left wheel torque inputs from the DC motors, r is the radius of the driving wheels, and l is the azimuth length from point C to the driving wheels.

Now $\bar{M}^{-1}$ is given by,

$$\bar{M}^{-1} = \begin{bmatrix} I & 0 \\ 0 & m \end{bmatrix} \tag{30}$$

and, the simplified dynamics of the mobile robot are given by,

$$\dot{V} = \begin{bmatrix} \dot{\nu}_c \\ \dot{\omega}_c \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & m \end{bmatrix} \frac{1}{r} \begin{bmatrix} 1 & 1 \\ l & l \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \tag{31}$$

$$\dot{V} = \begin{bmatrix} \dot{\nu}_c \\ \dot{\omega}_c \end{bmatrix} = \begin{bmatrix} \frac{I}{r} & \frac{I}{r} \\ \frac{lm}{r} & \frac{lm}{r} \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \tag{32}$$

2.3. Error Dynamics

The desired posture of the mobile robot, $P_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T$, is obtained from the inertial frame. The tracking error in the body-fixed frame is given by $e_P(t) = [e_D(t), e_L(t), e_\theta(t)]^T$, where e_D, e_L, and e_θ are the tracking errors in the driving, lateral, and orientation directions, respectively. These errors are calculated through a transformation matrix T_e:

$$e_P = T_e \begin{bmatrix} e_D \\ e_L \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \tag{33}$$

where $e_x = x_d - x_c$, $e_y = y_d - y_c$, and $e_\theta = \theta_d - \theta_c$ represent the tracking errors in the inertia frame.

The error dynamics are derived by taking the time derivative:

$$\begin{bmatrix} \dot{e}_D \\ \dot{e}_L \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \omega_c e_L - \nu_c + \nu_d \cos e_\theta \\ -\omega_c e_D + \nu_d \sin e_\theta \\ \omega_d - \omega_c \end{bmatrix} \tag{34}$$

where ν_d and ω_d are the linear and angular velocity of the mobile robot, respectively. Given the desired posture P_d, ν_d and ω_d are calculated by:

$$\nu_d = \sqrt{x_d^2 + y_d^2} \tag{35}$$

$$\omega_d = \frac{y_d'' \dot{x}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2} \tag{36}$$

2.4. KF AND UKF MODEL FOR THE MOBILE ROBOT

Based on equations (28)-(32) and using Euler approximation with a time step Δt, the KF state model for the dynamics of the mobile robot:

$$x_{k+1} = Ax_k + Buk + \alpha_k \tag{37}$$

is defined as:

$$\hat{V}_{k+1} = A\hat{V}_k + \bar{M}^{-1}\bar{B}\tau_{c,k} \cdot \Delta t + \alpha_k \tag{38}$$

$$\begin{bmatrix} \hat{v}_{c,k+1} \\ \hat{\omega}_{c,k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{v}_{c,k} \\ \hat{\omega}_{c,k} \end{bmatrix} + \begin{bmatrix} \frac{I}{r} & \frac{I}{r} \\ \frac{lm}{r} & \frac{lm}{r} \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \cdot \Delta t + \alpha_k \tag{39}$$

where $\hat{v}_{c,k}$ and $\hat{\omega}_{c,k}$ are the estimated linear and angular velocity at time k, and α_k is the system noise.

The measurement for the dynamics is calculated as:

$$z_{k+1} = Hx_{k+1} + \beta_{k+1} \tag{40}$$

$$\tilde{V}_{k+1} = H\hat{V}_{k+1} + \beta_{k+1} \tag{41}$$

$$\begin{bmatrix} \tilde{v}_{k+1} \\ \tilde{\omega}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{v}_{c,k+1} \\ \hat{\omega}_{c,k+1} \end{bmatrix} + \beta_{k+1} \tag{42}$$

Here, $\hat{V}_{k+1}$ and β_{k+1} are the measured velocity vector and the measurement noise at time $k+1$, respectively.

As for the UKF, the state and measurement model for the kinematics of the mobile robot is treated as:

$$\begin{bmatrix} \hat{x}_{c,k+1} \\ \hat{y}_{c,k+1} \\ \hat{\theta}_{c,k+1} \end{bmatrix} = \begin{bmatrix} \hat{x}_{c,k} + \cos(\hat{\theta}_{c,k})\hat{\upsilon}_{c,k}\Delta t \\ \hat{y}_{c,k} + \sin(\hat{\theta}_{c,k})\hat{\upsilon}_{c,k}\Delta t \\ \hat{\theta}_{c,k} + \hat{\omega}_{c,k}\Delta t \end{bmatrix} + \delta_k \tag{43}$$

$$\hat{P}_{k+1} = h(\hat{P}_{k+1}, \gamma_{k+1}) = \begin{bmatrix} \hat{x}_{c,k+1} \\ \hat{y}_{c,k+1} \\ \hat{\theta}_{c,k+1} \end{bmatrix} + \gamma_{k+1} \tag{44}$$

where $\hat{x}_{c,k}$, $\hat{y}_{c,k}$, and $\hat{\theta}_{c,k}$ are the estimates of the robot positions at time k, and δ_k and γ_k are the process and measurement noises, respectively. The UKF uses the unscented transform technique with 7 sigma points generated for the kinematic model.

2.5. Implementation of Backstepping Controller

The conventional backstepping tracking control law for a nonholonomic mobile robot is given as:

$$\upsilon_r = C_1 e_D + \upsilon_d \cos e_\theta \tag{45}$$

$$\omega_r = \omega_d + C_2 \upsilon_d e_L + C_3 \upsilon_d \sin e_\theta \tag{46}$$

where C_1, C_2, and C_3 are the designed parameters, υ_r and ω_r are respectively the reference linear and angular velocity commands generated by the controller. To address speed jump issues, Yang et al. proposed a bioinspired backstepping control based on the shunting model's neural dynamics equation:

$$C_m \frac{dV_m}{dt} = -(E_p + V_m)g_p + (E_{Na} - V_m)g_{Na} - (E_k + V_m)g_k \tag{47}$$

where C_m is the membrane capacitance, and E_p, E_{Na}, and E_k are Nernst potentials for passive leak, sodium ions, and potassium ions, respectively.

By setting $C_m = 1$ and substituting specific parameters, the shunting model is rewritten as:

$$\frac{d\xi}{dt} = -A_1\xi + (B_1 - \xi)S_+ - (D_1 + \xi)S_- \tag{48}$$

The proposed bioinspired backstepping control laws, incorporating the shunting model, are defined as:

$$\upsilon_r = \upsilon_s + \upsilon_d \cos e_\theta \tag{49}$$

$$\omega_r = \omega_d + C_2 \upsilon_d e_L + C_3 \upsilon_d \sin e_\theta \tag{50}$$

The neural dynamics equation for υ_s is given by:

$$\frac{d\upsilon_s}{dt} = -A_1\upsilon_s + (B_1 - \upsilon_s)f(e_D) - (D_1 + \upsilon_s)g(e_D) \tag{51}$$

where $f(e_D) = \max\{e_D, 0\}$ represents the linear above-threshold function, and $g(e_D) = \max\{-e_D, 0\}$ is a nonlinear function. The parameter A_1 denotes the passive decay rate, while B_1 and D_1 are the upper and lower bounds of the velocity.

The Lyapunov candidate function and its time derivative for bioinspired backstepping control are proposed as:

$$V_1 = \frac{1}{2}e_D^2 + \frac{1}{2}e_L^2 + \frac{1}{C_2}(1 - \cos e_\theta) + \frac{1}{2B_1}\upsilon_s^2 \tag{52}$$

$$\dot{V}_1 = \dot{e}_D e_D + \dot{e}_L e_L + \frac{1}{C_2}\dot{e}_\theta \sin e_\theta + \frac{1}{B_1}\dot{\upsilon}_s \upsilon_s \tag{53}$$

2.6. Implementation of Torque Controller

The actual velocity of the mobile robot differs from the velocity command generated by the bioinspired backstepping control due to noise. The velocity tracking error, denoted as e_η, is expressed as:

$$e_\eta = \begin{bmatrix} e_\upsilon \\ e_\omega \end{bmatrix} = \begin{bmatrix} \upsilon_r - \upsilon_c \\ \omega_r - \omega_c \end{bmatrix} \tag{54}$$

To mitigate this issue the paper comes with the torque tracking control law $\tau_c = [\tau_L, \tau_R]^T$ is proposed as:

$$\tau_L = \left(\frac{mr}{2}\right)(\dot{\upsilon}_r + C_4 e_\upsilon) - \left(\frac{Ir}{2c}\right)(\dot{\omega}_r + C_5 e_\omega) \tag{55}$$

$$\tau_R = \left(\frac{mr}{2}\right)(\dot{\upsilon}_r + C_4 e_\upsilon) + \left(\frac{Ir}{2c}\right)(\dot{\omega}_r + C_5 e_\omega) \tag{56}$$

where $C_4 = 5$, $C_5 = 5$, $c = 1$, I = 0.5 , r = 1, m = 10, and we get:

$$\tau_L = (5)(\dot{\upsilon}_r + 5\upsilon) - (0.05)(\dot{\omega}_r + 5\omega) \tag{57}$$

$$\tau_R = (5)(\dot{\upsilon}_r + 5\upsilon) + (0.05)(\dot{\omega}_+ 5\omega) \tag{58}$$

The Lyapunov candidate function and its time derivative for the torque controller can be defined as:

$$V_2 = V_1 + \frac{1}{2}e_\upsilon^2 + \frac{1}{2}e_\omega^2 \tag{59}$$

$$\dot{V}_2 = \dot{V}_1 + e_\upsilon(\dot{\upsilon}_r - \dot{\upsilon}_c) + e_\omega(\dot{\omega}_r - \dot{\omega}_c) \tag{60}$$

3. Stability Analysis

To establish the stability of the proposed control strategy, the asymptotic stability of the bioinspired backstepping controller and torque controller is proven. Subsequently, the overall stability is proven under the assumption that posture and velocity errors are bounded.

First, for the bioinspired backstepping controller, from the equations (49)-(51) and (53) $\dot{V}_1$ is rewritten as:

$$\dot{V}_1 = -\upsilon_s e_D - \frac{C_3}{C_2}\upsilon_d \sin^2 e\theta + \frac{1}{B_1}\left[-A_1 - f(e_D) - g(e_D)\right]\upsilon_s^2 + \frac{1}{B_1}\left[B_1 f(e_D) - D_1 g(e_D)\right]\upsilon_s. \tag{61}$$

Assuming $B_1 = D_1$, the expression becomes:

$$\dot{V}_1 = -\frac{C_3}{C_2}\upsilon_d \sin^2 e\theta + \frac{1}{B_1}\left[-A_1 - f(e_D) - g(e_D)\right]\upsilon_s^2 + \left[f(e_D) - g(e_D) - e_D\right]\upsilon_s \tag{62}$$

Based on the definitions of $f(e_D)$ and $g(e_D)$, if $e_D \geq 0$, then $f(e_D) = e_D$ and $g(e_D) = 0$, leading to:

$$[f(e_D) - g(e_D) - e_D]\upsilon_s = e_D - 0 - e_D = 0 \tag{63}$$

Applying the same concept when $e_D \leq 0$, where $g(e_D) = e_D$ and $f(e_D) = 0$, yields:

$$[f(e_D) - g(e_D) - e_D]\upsilon_s = 0 - (-e_D) - e_D = 0 \tag{64}$$

Thus, (62) becomes:

$$\dot{V}_1 = -\frac{C_3}{C_2}\upsilon_d \sin^2 e\theta + \frac{1}{B_1}[-A_1 - f(e_D) - g(e_D)]\upsilon_s^2 \tag{65}$$

Given that C_3, C_2, and υ_d are positive constants, and $g(e_D)$ and $f(e_D)$ are non-negative, while A_1 and B_1 are non-negative constants, it follows that $\dot{V}_1 \leq 0$. The torque controller is asymptotically stable only when $e_D = 0$ and $e\theta = 0$.

To prove the stability of the torque controller, integrating (28) into (60) results in the equation:

$$\dot{V}_2 = \dot{V}_1 + e\upsilon(\dot{\upsilon}_r - \frac{1}{m_r}\tau_L - \frac{1}{m_r}\tau_R) + e\omega(\dot{\omega}_r + \frac{l}{I_r}\tau_L - \frac{l}{I_r}\tau_R) \tag{66}$$

By replacing τ_L and τ_R in (49) and (50) into (58), $\dot{V}_2$ is rewritten as:

$$\dot{V}_2 = \dot{V}_1 - C_4 e\upsilon^2 - C_5 e\omega^2 \tag{67}$$

Since $\dot{V}_1$, $-C_4 e\upsilon^2$, and $-C_5 e\omega^2$ are all less than or equal to zero, the proposed torque controller reaches a stable condition. When e_D, $e\theta$, and $e\eta$ are zeros, $\dot{V}_2 = 0$, establishing the asymptotic stability of the torque control.

For the overall stability analysis, the Lyapunov candidate function of the system is defined as:

$$V_3 = V_1 + V_2 \tag{68}$$

The time derivative of V_3 is derived as:

$$\dot{V}_3 = 2\left(-\frac{C_3}{C_2}\upsilon_d \sin^2 e\theta + \frac{1}{B_1}\left[-A_1 - f(e_D) - g(e_D)\right]\upsilon_s\right) - C_4 e\upsilon^2 - C_5 e\omega^2 \tag{69}$$

When $e\theta = [-\pi, \pi]$, the control system is globally asymptotically.

4. Results

In this study, researchers developed a sophisticated tracking control approach for robotic systems, tackling issues including dynamic constraint consideration, resilience in the presence of noise, and smooth velocity instructions. Our implementation made use of the methods in the paper which were the Unscented Kalman Filter (UKF), a torque controller, and a backstepping controller with biological inspiration.

We carried out our implementation in Python and implemented Kalman Filter from scratch and therefore the results that we received had some discrepancies. The strategies proposed by this paper effectively reduced the velocity jumps and overshoots that come with traditional backstepping control and the torque controller allowed the robotic system to be controlled more precisely and smoothly. By guaranteeing precise state estimations, the integration of UKF and KF improved the control strategy even further.

After conducting a thorough implementation and investigation, we found that the suggested control approach not only produced precise state estimations but also successfully handled the difficulties caused by significant velocity jumps. Even though we implemented the Kalman Filter from scratch, the suggested control approach gave astounding results on a not so robust code.

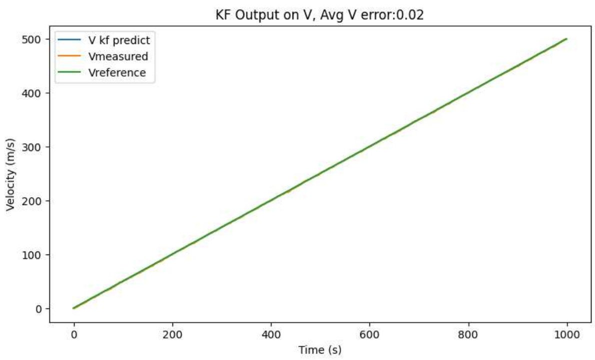

Fig. 1. Kalman Filter values on Velocity

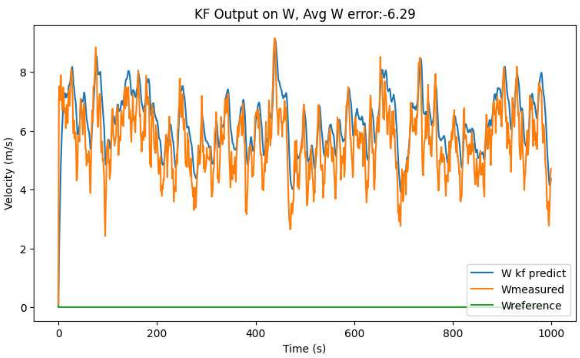

Fig. 2. Kalman Filter values on Angular Velocity

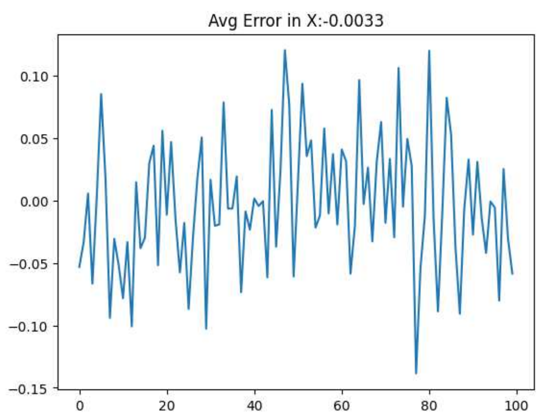

Fig. 3. X position Error - UKF

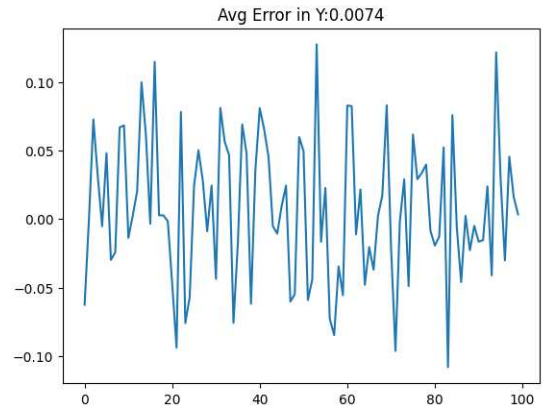

Fig. 4. Y position Error- UKF

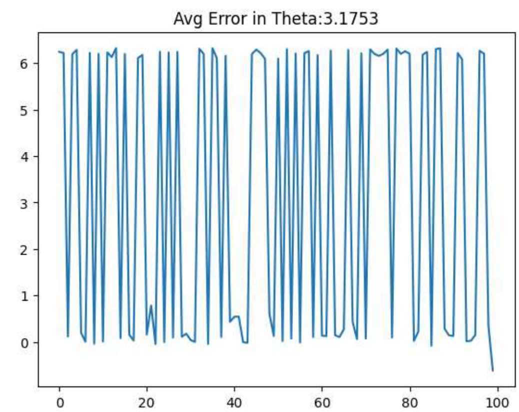

Fig. 5. Theta Error - UKF

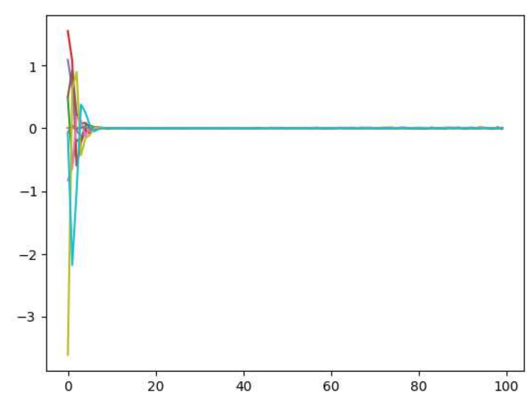

Fig. 6. UKF+KF

5. Conclusions

In this paper, a novel tracking control strategy featuring an enhanced bioinspired backstepping controller, comprising the bioinspired backstepping controller and the torque controller, has been introduced. This method demonstrates superior trajectory tracking accuracy by considering both the kinematics and dynamics of the mobile robot. Through various experiments and analyses presented in this study, it is evident that the proposed control strategy excels in providing smooth velocity commands while reducing the impact of external noises.

To enhance the real-world applicability of the proposed approach, Kalman Filter (KF) and Unscented Kalman Filter (UKF) are integrated with the torque controller and bioinspired backstepping controller, respectively. This integration contributes to improved performance in state estimates, making the system more robust and adaptable to dynamic and noisy environments. The asymptotic stability of the entire closed-loop system has been proven using Lyapunov stability theory.

In our implementation, we translated the theoretical foundations outlined in the paper into a practical solution using the Python programming language. By implementing the bioinspired backstepping controller, torque controller, and the Kalman filters in Python, we successfully replicated and validated the proposed control strategy. The implementation not only reinforced the theoretical findings of the paper but also provided a tangible and effective solution for autonomous mobile robots operating under challenging conditions with system and measurement noises.

In conclusion, this paper contributes a valuable and practical solution to the field of mobile robot trajectory tracking, showcasing the effectiveness of the enhanced bioinspired backstepping controller in real-world scenarios. The successful implementation in Python further underscores the feasibility and adaptability of the proposed control strategy for practical applications.

6. Bibliography

[1]Z. Xu, S. X. Yang and S. A. Gadsden, ”Enhanced Bioinspired Backstepping Control for a Mobile Robot With Unscented Kalman Filter,” in IEEE Access, vol. 8, pp. 125899-125908, 2020, doi: 10.1109/ACCESS.2020.3007881.
[2]https://jgoslinski.medium.com/the-unscented-kalman-filter-simply-the-best-python-code-5cd5ebaebf5f
[3]https://www.youtube.com/watch?v=X42HqGthOqslist=PLvKAPIGzFEr8n7WRx8RptZmC1rXeTzYtA