Estimation for slip ratio and adaptive control of Tracked mobile robot

Juo-Tung Chen and Jing-Shiang Wu

***Abstract*— when it comes to controlling a tracked mobile robot, the longitudinal slippage of the left and right tracks can be described by two unknown parameters. In this project, we intend to use different adaptive laws and neural network methods to estimate these slippage parameters. To compare the performance of the different estimation methods, we use Simulink and python to construct simulation models. An empirical formula of slippage was used to simulate slippage. The final results are analyzed by the the RMS value of the trajectory tracking error.**

I. INTRODUCTION

Recently, different applications of tracked mobile robots (TMR) are emerging more and more often in the fields of agriculture, national defense, and industry. As shown in Fig. 1, large contact areas between tracks and the ground provides better mobility in unconstructed environments. Subsequently, the problem of motion control of mobile robots becomes a popular topic among researchers. Lots of researchers have designed various tracking controllers [4] [9] [12] with the nonholonomic constraints, which is the assumption that the mobile robot are subject to a "pure-rolling without slipping" situation. However, the disadvantage of the tracks is that it increases the chance of slippage to happen in a rough terrain. Therefore, it is necessary to consider the slipping condition for mobile robots operating in complex environments. In [13], the slippage for mobile robot was categorized into lateral and longitudinal slippage. The trajectory tracking problem considering slippage was proposed in [2] and [10]. However, these studies all assume that the slippage parameters can be directly detected by the sensors in real time, which can still hardly be achieved by modern practical engineering. Hence, to compensate for the slippage, the slippage parameters must be estimated. There are many existing researches trying to solve this problem using different approaches. The first popular approach is using adaptive law to estimate the slipping parameters [1] [7]. On the other hand, neural network are often well suited to model uncertain or nonlinear function such as mobile robot dynamics [5]. Assumed that the slippage ratio can be described by a complicated nonlinear function g(x). This time, we try to approximate the g(x) with NN.

II. PROBLEM FORMULATION

Slippage ratio is defined as Eq.1

$$i_L = \frac{r\omega_L - v_L^s}{r\omega_L}, i_R = \frac{r\omega_R - v_R^s}{r\omega_R} \tag{1}$$

Where ω_L and ω_R are the angular velocities of the left and right wheel, r is the radius of the wheels, and v_L^s and v_R^s are the actual linear velocities of the wheel with slippage. Since

Fig. 1: The tracked mobile robot

ω_L and ω_R can be measured by the encoder or any other angular velocity sensors, if we can obtain the information of v_L^s and v_R^s, then the slippage ratio can be directly calculated. However, there are several difficulties that impede us from doing so. First, the nonlinear nature of the slippage parameters makes it difficult to measure the actual linear velocities of the wheel. Even if there are sensors like GPS that can obtain the information with relatively high precision, the cost of these sensors is usually unaffordable. However, using low-cost sensors to measure the linear velocities can result in magnificent noise in the signal. In addition, the low-cost sensors usually provide low sampling rate and delay, which is also a big problem when controlling the robot. Therefore, using limited information to design an adaptive law which can indirectly estimate the slippage ratio becomes a crucial topic regarding the precise control of mobile robots.

In this project, we will use different approaches to estimate the slippage ratio and compare the trajectory tracking performance.

III. APPROACHES

The kinematics of the mobile robot is defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{b} & \frac{-1}{b} \end{bmatrix} \begin{bmatrix} \omega_L(1-i_L) \\ \omega_R(1-i_R) \end{bmatrix}$$

where $q = [x, y, \theta]^T$ is the Cartesian coordinate of the mobile robot, and $\dot{q} = [\dot{x}, \dot{y}, \omega]^T$ is the time derivative of the position of the robot. Also, r is the radius of the wheel, b is the distance between two driving wheels of the TMR. ω_L, ω_R are the angular velocities of the left and right driving wheels, which is controllable.

To achieve trajectory tracking, we define the tracking error $e = [e_1, e_2, e_3]^T$ in the global Cartesian coordinate as:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ x_r - y \\ \theta_r - \theta \end{bmatrix}$$

By applying backstepping control method, the auxiliary control law can be design as [3]:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_R + k_2 v_r e_2 + k_3 \sin e_3 \end{bmatrix}$$

Therefore, the actual control input will be:

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \dfrac{1}{1-i_L} & \dfrac{-b}{2(1-i_L)} \\ \dfrac{1}{1-i_R} & \dfrac{b}{2(1-i_R)} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

To calculate for the control input, we need the information of the slippage parameters i_L, i_R

A. Adaptive update law 1

For the first adaptive law, we adopted the method proposed in [1]. The adaptive law designed in this research doesn't estimate the slippage parameters directly. Instead, the parameters were redefined as following:

$$a_k = \frac{1}{1-i_k}, k = L, R \tag{2}$$

The estimations of a_k are defined as $\hat{a}_k$. On the other hand, the estimation error is defined as $\bar{a}_k = \hat{a}_k - a_k$.

The Lyapunov function was chosen to be as following:

$$V(t) = \frac{1}{2}(e_1^2 + e_2^2) + \frac{1-\cos e_3}{k_2} + \frac{\bar{a_L}}{2\rho_1 a_L} + \frac{\bar{a_R}}{2\rho_2 a_R} \tag{3}$$

where k_2, ρ_1, ρ_2 are positive constants.

Therefore, the adaptive law for a_L and a_R can be designed as Eq. 4 and Eq. 5 to make $\dot{V}(t) \leq 0$:

$$\dot{\hat{a}}_L = \rho_1[(\frac{e_2}{b}+\frac{1}{2})e_1 v - (\frac{e_2}{2}+\frac{b}{4})e_1\omega - \frac{\sin e_3}{k_2}(\frac{v}{b}-\frac{\omega}{2})] \tag{4}$$

$$\dot{\hat{a}}_R = \rho_2[-(\frac{e_2}{b}-\frac{1}{2})e_1 v - (\frac{e_2}{2}-\frac{b}{4})e_1\omega + \frac{\sin e_3}{k_2}(\frac{v}{b}+\frac{\omega}{2})] \tag{5}$$

B. Adaptive update law 2

For the second adaptive law, the method proposed in [7] was adopted. Similarly, the adaptive law designed in this research estimates the parameters defined as Eq. 2

The difference is that the Lyapunov function was chosen to be as following:

$$V(e) = V_0(e) + \frac{\bar{a_L}^2}{2\gamma_1 a_L} + \frac{\bar{a_R}^2}{2\gamma_2 a_R} \tag{6}$$

where k_2, ρ_1, ρ_2 are positive constants, and $V_0(e)$ is the Lyapunov function of the mobile robot tracking controller [8].

Therefore, the adaptive law for a_L and a_R can be designed as Eq. 7 and Eq. 8 to make $\dot{V}(t) \leq 0$:

$$\begin{aligned} \dot{\hat{a}}_L = \gamma_1(v - \frac{b}{2}\omega)[(\frac{e_2}{b}+\frac{1}{2})e_1 \\ - (e_2 + k_3 e_3)\frac{e_1}{b} - \frac{k_3(e_2+k_3e_3)}{b} - \frac{\sin e_3}{bk_2}] \end{aligned} \tag{7}$$

$$\begin{aligned} \dot{\hat{a}}_R = \gamma_2(v + \frac{b}{2}\omega)[-(\frac{e_2}{b}-\frac{1}{2})e_1 \\ + (e_2 + k_3 e_3)\frac{e_1}{b} + \frac{k_3(e_2+k_3e_3)}{b} + \frac{\sin e_3}{bk_2}] \end{aligned} \tag{8}$$

C. Neural Network

We have introduced the application of neural network. In this paper, we try to approximate the nonlinear function g(x) and input is defined as

$$\theta = \begin{bmatrix} w_{Rref}(i-1) & \dot{w_{Rref}}(i-1) & w_R(i-1) \\ w_{Lref}(i-1) & \dot{w_{Lref}}(i-1) & w_L(i-1) \end{bmatrix}^T$$

The neural network can be redefined as $g(\theta) : R^6 \to R^2$

$$g(\theta) = W^T \Phi(\theta) \tag{9}$$

where Φ is activation function, W are weights in neural network and g is regarded as the output predicted slippage ratio $[i_L(i) \quad i_R(i)]^T$. Next we choose MLP with 6 inputs, 1 hidden layer with 10 neuron and 2 outputs, all of the layers takes the sigmoid activation function as our NN structure. Introduce the SGD as the optimizer and adopt MSE as loss function. In order to get a better performance, we train this network in the following trajectory:

$$\begin{cases} x = 0.01t \\ y = (0.01t)^2 \qquad (0 \leq t < 200) \end{cases}$$

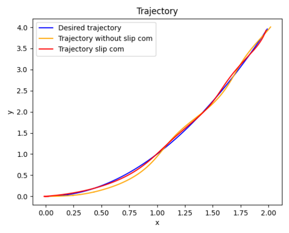
training with epochs = 200, datasets = 200 and batches = 32. Fig. 2 shows the training result.

Next, we will apply this pretrained model to a different trajectory and validate it if it can accurately predict the slippage ratio. Moreover, we also introduce the online learning [6] to compare the tracking performance and slippage ratio estimation with the method of offline pretrained model.

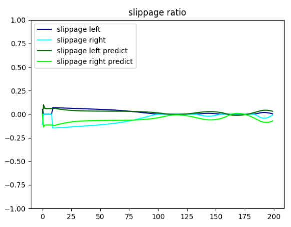
IV. SIMULATION AND DISCUSSION

After the methods have been determined, we use simulation to compare the performance for each method. To model the behavior of the vehicle, some parameters must be assumed in the simulation. The parameters of the TMR are set as follows: $b = 0.15m$, $r = 0.075m$, $\zeta = 1$, $k_2 = 14, k_1 = k_3 = 1$.

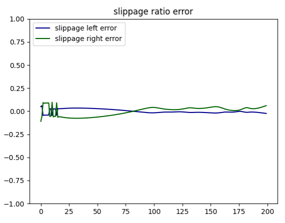
In addition, since slippage is a nonlinear behavior, using a constant to analyze the estimation performance cannot guarantee an equivalent result when implementing on the robot. Therefore, we adopted an empirical formula proposed by [11] to simulate the slippage. According to the research,



(a) training tracking result



(b) slippage estimation



(c) slippage error

Fig. 2: Training result

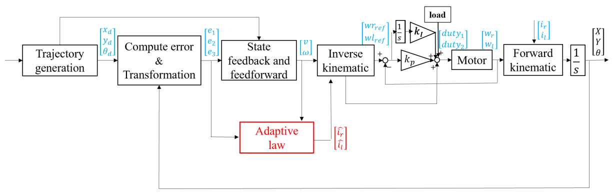


Fig. 3: The block diagram of the trajectory tracking scheme

the slippage parameters are exponential functions of the turning radius, which can be written as follows:

$$\begin{cases} i_L = 0.07e^{-0.68R} \\ i_R = -0.15e^{-0.63R} \end{cases} \quad \dot{\theta} \leq 0$$

$$\begin{cases} i_L = -0.15e^{-0.63R} \\ i_R = 0.07e^{-0.68R} \end{cases} \quad \dot{\theta} \geq 0$$

The coefficients in the formula are obtained from the average of multiple experiments. R is the radius of path and can be calculated by

$$R = \frac{L(w_R + w_L)}{(w_R - w_L)} \qquad for \quad w_r > w_l$$

Also, to verify the effect of the estimation schemes, we increase the values of the slippage parameters by 1.3 times as much as the previous values at t = 30 sec.

The block diagram of the entire trajectory tracking scheme is shown in Fig3.

The equation of the reference trajectory is given as:

$$\begin{cases} x = 1.6\sin(0.01t) \\ y = -1.6\cos(0.01t) \qquad (0 \leq t < 81) \end{cases}$$

where $t \geq 0$ is the simulation time. The initial posture of the reference trajectory is set at:

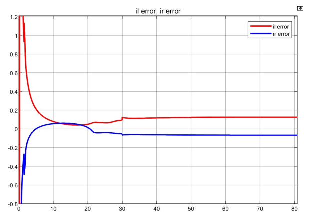
$$q_R(0) = [x_R(0), y_R(0), \theta_R(0)]^T = [0, 0, 0]^T$$

However, to demonstrate the tracking ability, the actual initial posture of the TMR is set at:

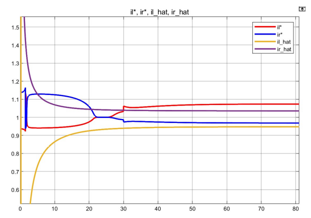
$$q(0) = [x(0), y(0), \theta(0)]^T = [0.1, -0.1, \frac{1}{4}\pi]^T$$

A. Adaptive laws

The initial conditions are chosen as $\hat{a}_L(0) = \hat{a}_R(0) = 1$. Also, the estimation gains are set to be $\rho_1 = \rho_2 = 5$, $\gamma_1 = \gamma_2 = 50$. The results of the first adaptive law can be seen in Fig. 4



(a) The estimation errors

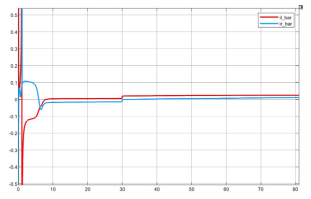


(b) slippage estimations

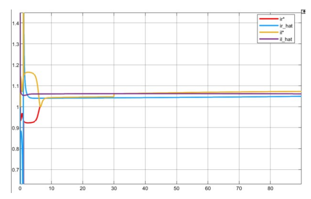
Fig. 4: The estimation results of the first adaptive law

We can see that the estimation errors of the parameters quickly reduce to about 0.04 within 10 secs. However, with the change of the angular velocity of the two wheels, the slippage ratios also change. The convergence of the estimation couldn't keep up with the sudden change of the parameters, resulting in a rising of the estimation errors. Moreover, when the slippage ratio suddenly magnifies at $t = 30$, the estimation errors rise again. The final estimation error when $t = 81$ is $\bar{i}_L = 0.1253$, $\bar{i}_R = 0.0669$

For the second adaptive law we adopted, the results are in Fig.5



(a) The estimation errors



(b) slippage estimations

Fig. 5: The estimation results of the second adaptive law

From the results, we can see the convergence happens faster than the first adaptive law. The estimation errors reduced to $<$ 0.001 within 8 secs. However, when the parameters increase at $t = 30$, the changes in the estimations are inapparent. Nonetheless, the estimation performance is still better than the first adaptive law. The final estimation error when $t = 81$ is $\bar{i}_L = 0.02398$, $\bar{i}_R = 0.01026$

Fig. 6 demonstrates the trajectory tracking results of the TMR using the two different adaptive laws. To distinguish the effect after compensating for the slippage, the tracking result without using the adaptive controller was also in the figure.

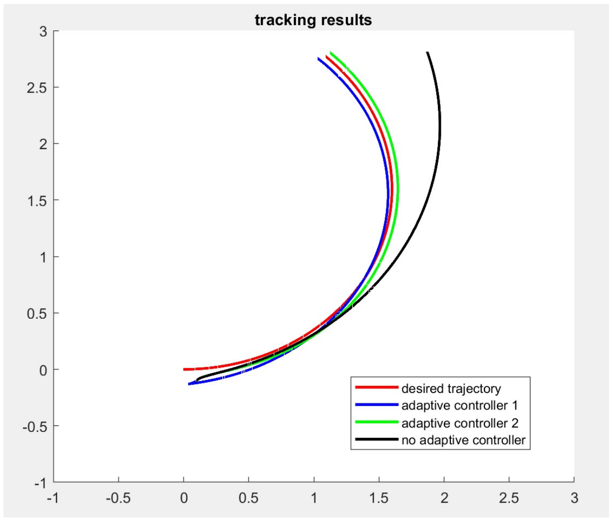


Fig. 6: The trajectory tracking result with the two adaptive laws

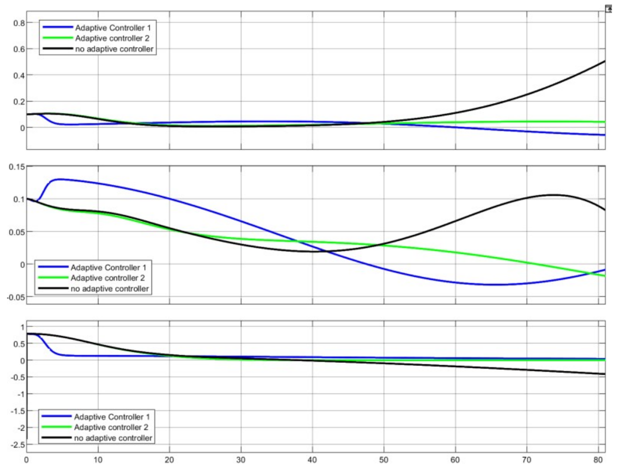


Fig. 7: The tracking errors with the two adaptive laws

From the tracking result, it is clear that with the implementation of adaptive controller, the negative effect of the slippage can be canceled, which greatly benefits the tracking performance. On the other hand, the difference between the two trajectories using different adaptive laws are not very obvious. Fig. 7 shows the tracking error along the reference trajectory.

Since the convergence rate using the first adaptive law is slower, the result using it has more error at the beginning. Therefore, despite the insignificant difference, we can still suggest that the tracking ability using the second adaptive law are slightly better.

B. Neural Network

We choose python and ML tool box, keras as our simulation platform to construct our model.

1) The slippage ratio remains the same: Fig.8 the result shows that the tracking performance with the offline NN slippage compensator is better than that without the compensator. However, because the NN training data is not generated

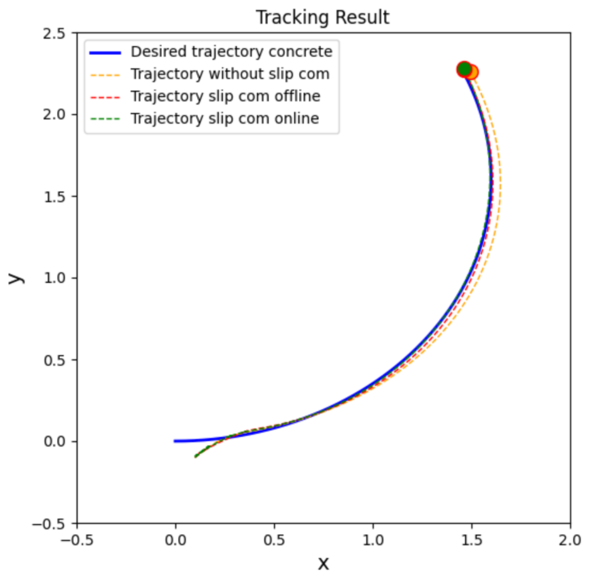


Fig. 8: Tracking result(slippage not changes)

from this desired trajectory, the offline tracking trajectory is not as good as the online tracking trajectory.

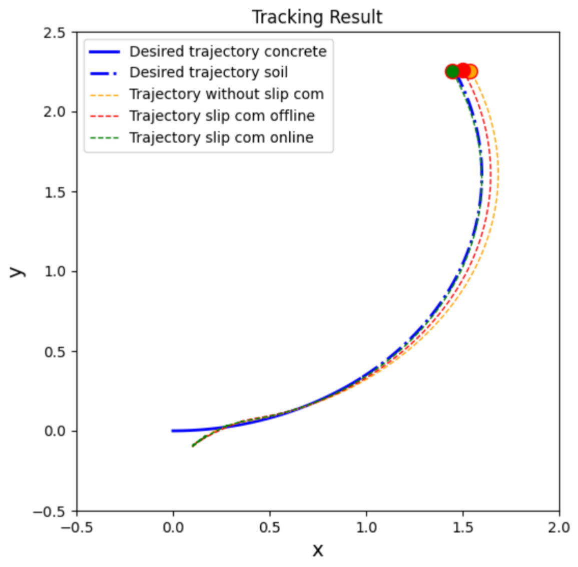
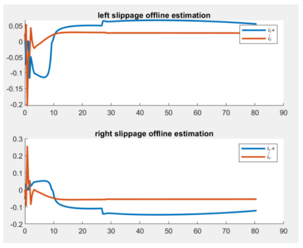


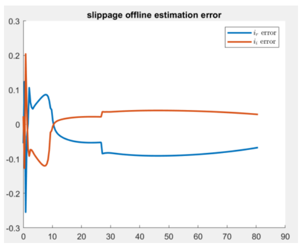
Fig. 9: Tracking result(slippage changes when t > 30sec)

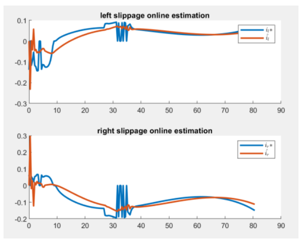
2) The slippage ratio changes when t > 30sec: In this case, the defect of the offline NN slippage compensator reveals, which is that us cannot catch the desired trajectory. By contrast, tracking with online NN compensator almost overlaps with the desired trajectory.(Fig.9)

Fig.10 shows the comparison of slippage ratio estimation and estimation error between online and offline NN compensator. Online NN compensator has smaller slippage ratio error and better slippage tracking performance.

3) offline v.s. online: We can regard the changing in the slippage ratio as the different terrain or soil condition the TMR robot encounters in the real world. A lot of factors


(a) slippage estimation offline


(b) slippage error offline


(c) slippage estimation online

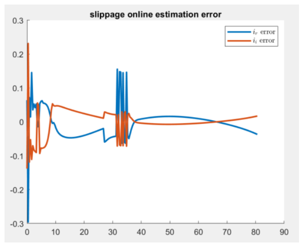

(d) slippage error online

Fig. 10: NN offline/online slippage estimation result

that affect the slippage ratio we won't know in the future. Therefore we need online learning to update the slippage compensator model in time.

C. NN v.s. Adaptive

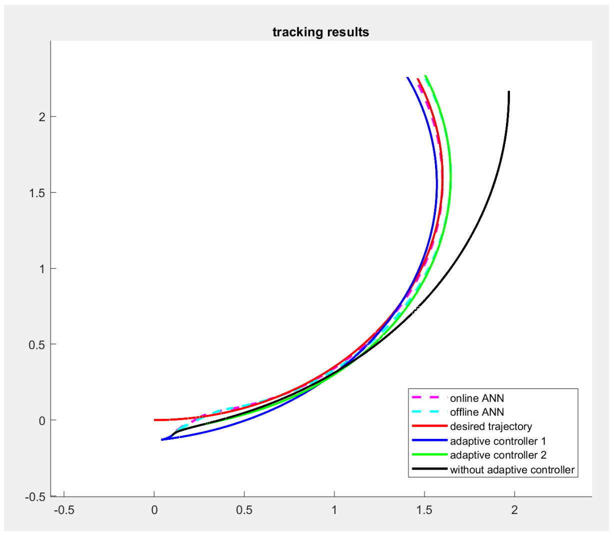


Fig. 11: Tracking result of four compensator methods

Fig.11 shows the tracking result of the four compensator methods. All of the four methods effectively predict the slippage ratio. To evaluate which method come to a best tracking performance. We introduce RMS value of tracking error as criterion

$$|e| = \sqrt{\frac{e_x^2 + e_y^2 + e_\theta^2}{3}} \tag{10}$$

we define

$$e_x = \int |x_{error}| dt$$
$$e_y = \int |y_{error}| dt$$
$$e_\theta = \int |\theta_{error}| dt$$

Fig. 12 shows the error of x, y and θ along the tracking process and Table I shows the RMS value of four slippage compensator methods.

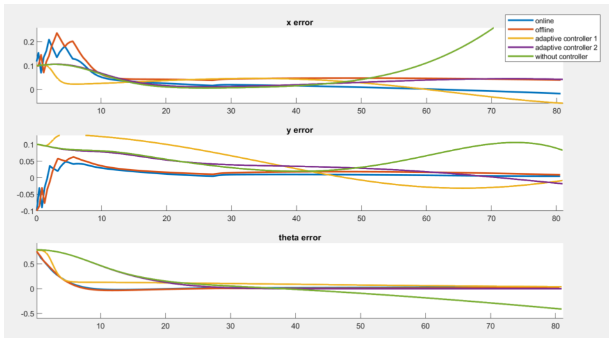


Fig. 12: error data

	e_x	e_y	e_θ	$\|e\|$
Adaptive 1	3.341	4.525	9.188	6.219
Adaptive 2	3.471	3.315	10.180	6.500
NN online	5.824	2.571	9.340	6.525
NN offline	11.558	4.105	7.895	8.435

TABLE I: RMS value of four slippage compensator

Only offline NN compensator has larger RMS, and the other three compensators have outstanding performance.

V. CONCLUSIONS

In this paper, we describe how difficult to model the slippage ratio and how slippage ratio have an impact on tracking trajectory. To deal with the problem of slippage ratio, we introduce two methods of adaptive designs and online/offline neural network to estimate the slippage ratio. Finally, we prove that we can predict the slippage ratio correctly and enhance tracking performance by matlab and python simulation. Next, we simulate these slippage-ratio-predicted methods on various trajectories to validate its robustness. Finally, we will implement it on a real TMR robot.

CONTRIBUTION

In this final project, the overall contributions are equally distributed. Juo-Tung chen is responsible for the adaptive law simulation using Simulink, whereas Jing-Shiang Wu is responsible for the Neural Network implementation using Python.

For the composition of the paper, Juo-Tung is responsible for Abstract, Introduction, and Problem formulation. On the other hand, Jing-Shiang writes about the Results and Conclusions.

References

[1] Mingyue Cui. Observer-based adaptive tracking control of wheeled mobile robots with unknown slipping parameters. *IEEE Access*, 7:169646–169655, 2019.

[2] Mingyue Cui, Rongjie Huang, Hongzhao Liu, Xuyan Liu, and Dihua Sun. Adaptive tracking control of wheeled mobile robots with unknown longitudinal and lateral slipping parameters. *Nonlinear Dynamics*, 78(3):1811–1826, 2014.

[3] Mingyue Cui, Hongzhao Liu, Wei Liu, and Yi Qin. An adaptive unscented kalman filter-based controller for simultaneous obstacle avoidance and tracking of wheeled mobile robots with unknown slipping parameters. *Journal of Intelligent & Robotic Systems*, 92(3):489–504, 2018.

[4] Mingyue Cui, Wei Liu, Hongzhao Liu, Hualong Jiang, and Zhipeng Wang. Extended state observer-based adaptive sliding mode control of differential-driving mobile robot with uncertainties. *Nonlinear Dynamics*, 83(1-2):667–683, 2016.

[5] Haibo Gao, Xingguo Song, Liang Ding, Kerui Xia, Nan Li, and Zongquan Deng. Adaptive motion control of wheeled mobile robot with unknown slippage. *International Journal of Control*, 87(8):1513–1522, 2014.

[6] Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey, 2018.

[7] Juliano G Iossaqui, Juan F Camino, and Douglas E Zampieri. A nonlinear control design for tracked robots with longitudinal slip. *IFAC Proceedings Volumes*, 44(1):5932–5937, 2011.

[8] Doh-Hyun Kim and Jun-Ho Oh. Globally asymptotically stable tracking control of mobile robots. In *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, volume 2, pages 1297–1301. IEEE, 1998.

[9] Yongfu Li, Chuancong Tang, Srinivas Peeta, and Yibing Wang. Integral-sliding-mode braking control for a connected vehicle platoon: theory and application. *IEEE Transactions on Industrial Electronics*, 66(6):4618–4628, 2018.

[10] Chang Boon Low and Danwei Wang. Gps-based path following control for a car-like wheeled mobile robot with skidding and slipping. *IEEE Transactions on control systems technology*, 16(2):340–347, 2008.

[11] S Ali A Moosavian and Arash Kalantari. Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 95–100. IEEE, 2008.

[12] Luy Nguyen Tan. Omnidirectional-vision-based distributed optimal tracking control for mobile multirobot systems with kinematic and dynamic disturbance rejection. *IEEE Transactions on Industrial Electronics*, 65(7):5693–5703, 2017.

[13] Danwei Wang and Chang Boon Low. Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective. *IEEE Transactions on Robotics*, 24(3):676–687, 2008.