Vector-Based Modeling and Trajectory Tracking Control of Autonomous Two-Wheeled Robot

Nur Uddin*, *Member, IAENG,* Hendra G. Harno, and Wahyu Caesarendra

***Abstract*—This paper presents a new approach to kinematic modeling and trajectory tracking control system (TTCS) design of an autonomous two-wheeled robot. The robot kinematic modeling is carried out based on a vector diagram. It transforms the robot trajectory tracking problem into a stabilization problem of non-linear posture error dynamics. A solution is presented by applying a state feedback control. The posture error dynamics is linearized using the Taylor series and linear quadratic regulator (LQR) is applied to design the state feedback control. It results in a closed-loop system where all of the eigenvalues are real negative numbers. Performance evaluation of the TTCS is presented through numerical simulations in computer. The results show that the robot is able to track a reference trajectory without oscillation regardless of the robot initial posture.**

***Index Terms*—autonomous robot, kinematics modeling, trajectory tracking control, optimal control design.**

I. Introduction

AUTONOMOUS mobile robot has capability to move from a departure point to a destination point through a desired route without any intervention of the robot operator. It is an enhancement of conventional mobile robot by applying a control system for steering the robot. This control system is called as the trajectory tracking control system (TTCS). The TTCS works to steer the robot track a reference trajectory by utilizing feedback signals from navigation sensors.

The autonomous mobile robots are one of the challenging research topics since the last three decades [1]–[3]. Recently, this topic gets more attention due to the emerging of advanced technology in electronics and computer systems. A great extent of efforts and works have been invested by researchers for developing autonomous robots [4]. The autonomous robots are not only provides challenges on the control and navigation systems but also on applying the latest technology such as artificial intelligence and computer vision [5]–[7].

There are several types of mobile robots including: aerial robot, water surface robot, underwater robot and ground mobile robot. A two-wheeled robot (TWR) is a part of the ground mobile robot where the robot's body is supported by two wheels only. The TWR is an interesting robot because the two-wheel support renders the robot agile with high maneuverability. Unfortunately, the TWR is unable to stand and categorized as a statically unstable system. A state feedback control system is applied to actively stabilize the TWR such that the robot is balanced. Several results on the TWR active stabilization have been presented based on various methods [8]–[10]. The balanced TWR is therefore ready to be deployed as an autonomous mobile robot with high maneuverability.

The first autonomous two-wheeled robot (ATWR) was successfully developed by Koyanogi et al. [11]. They designed a two-dimensional trajectory control system for a wheeled inverted pendulum robot. The wheeled inverted pendulum robot is similar to the TWR. Their work resulted in the robot moving autonomously. However, the robot can only be operated for low-speed maneuvers. Another ATWR that was able to maneuver at higher speed was then presented in [2], [3]. Control system of the robot was designed by including three tasks: 1) balancing and velocity control, 2) heading control, and 3) trajectory tracking control. The experimental test confirmed a capability of the robot to move autonomously up to approximately 20 cm/s.

Since then, several works on developing ATWR have been reported. Control system of the ATWRs were mostly designed using the model-based control method, for examples in [12]–[16]. Applying the model-based control method requires a dynamic model of the robot in deriving a control law. The model used is described by a set of mathematical equations representing the robot dynamics obtained through a modeling process. It can be distinguished in the literature that there are three underlying methods to perform TWR modeling [17] that include the Newtonian method [18]–[20], the Lagrangian method [3], [21]–[23], and the Kane's method [24]–[26]. Modeling the TWR is a relatively complex task because the TWR is a three-dimensional system with longitudinal and lateral motions. In this regard, the longitudinal motion is when the robot performs a pitching motion for robot balancing. Moreover, the lateral motion is when the robot performs a yawing motion for robot steering.

Considering the importance of the TWR modeling, we hence present as the main contribution of this paper a new approach to modeling the TWR's lateral motion based on a vector diagram. The main advantage of applying the vector diagram is that it gives a clear description on how to derive the robot's kinematic model. The resulting model is then used for designing a TTCS to yield the ATWR. Presentation of this paper is then organized as follows. The kinematic modeling of the robot based on the vector diagram is described in Section II. The resulted model is then applied in designing a TTCS for the TWR by using the LQR method as discussed in Section III. Performance evaluation of the designed TTCS is presented in Section IV. Numerical simulations are carried out to demonstrated performance of the TTCS applied in a TWR to track a reference trajectory. The performance

Manuscript received April 19, 2021; revised August 08, 2021.

N. Uddin is an Assistant Professor at the Department of Informatics and a research member of the Center for Urban Studies, Universitas Pembangunan Jaya, Tangerang Selatan, Indonesia (*Corresponding author, e-mail: nur.uddin@upj.ac.id).

H.G. Harno is a Senior Researcher at the Department of Aerospace and Software Engineering, Gyeongsang National University, Jinju, Republic of Korea (e-mail: h.g.harno@gmail.com).

W. Caesarendra is an Assistant Professor at the Faculty of Integrated Technologies, Universiti Brunei Darussalam, Gadong, Brunei (e-mail: wahyu.caesarendra@ubd.edu.bn).

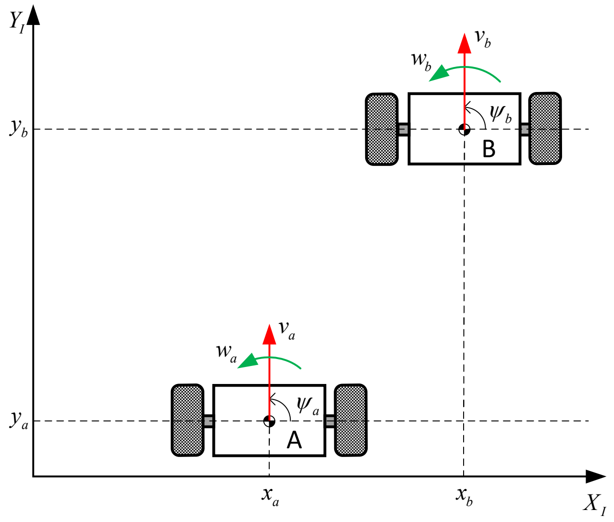

Fig. 1. Position and orientation of the TWR A and TWR B on the planar space.

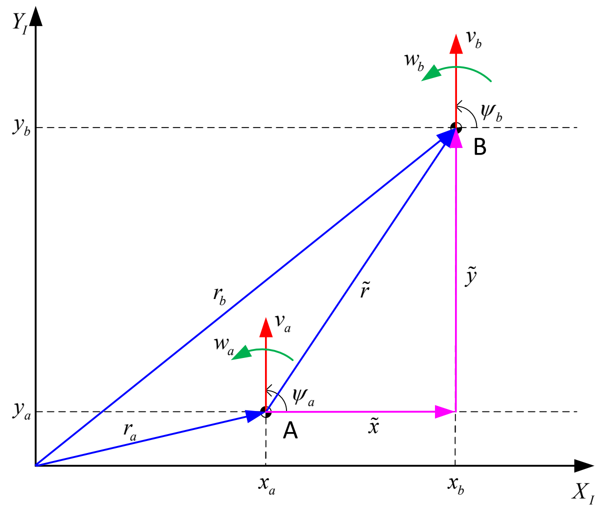

Fig. 2. A vector diagram of the TWR-A and TWR-B positions with respect to the inertial coordinate system.

evaluation is done through analysing the simulation results. Finally, conclusions of this work are given in Section V.

II. Vector-Based Modeling of the Robot Kinematics

Fig. 1 shows two units of TWRs on a planar space, which are named as the TWR-A and TWR-B, respectively. Each TWR can perform two kinds of movement on the planar space, i.e. translation and rotation. Position and orientation of the robots are presented in an inertial coordinate system $X_I Y_I Z_I$. The inertial coordinate system is a fixed frame coordinate system and shown by the X_I and Y_I axes in the Fig. 1. The Z_I axis is not appear in the figure as it is pointing out of the figure.

Positions of the TWR-A and TWR-B are given by (x_a, y_a) and (x_b, y_b), respectively. Meanwhile, orientations of the TWR-A and TWR-B are expressed by ψ_a and ψ_b, respectively. The TWR orientation shows a forward-movement direction of the robot. This orientation is represented by an angle of the robot's linear velocity with respect to the X_I axis.

Position and orientation of a robot are together referred to as a posture. Therefore, postures of both TWRs can be defined as follows:

$$\xi_a = \begin{bmatrix} x_a \\ y_a \\ \psi_a \end{bmatrix} \text{ and } \xi_b = \begin{bmatrix} x_b \\ y_b \\ \psi_b \end{bmatrix}. \quad (1)$$

where ξ_a is the posture of TWR-A and ξ_b is the posture of TWR-B.

When the TWR moves on the planar space, its posture always changes over time. Rate of the posture change is a time derivative of the posture which is a function of the robot velocities. Rate of change of the TWR-A posture is given as follows:

$$\begin{aligned} \dot{x}_a &= v_a \cos \psi_a, \\ \dot{y}_a &= v_a \sin \psi_a, \\ \dot{\psi}_a &= w_a, \end{aligned} \quad (2)$$

where v_a is the linear velocity of the TWR-A and w_a is the angular velocity of the TWR-A. Meanwhile, rate of change of the TWR-B posture is given by:

$$\begin{aligned} \dot{x}_b &= v_b \cos \psi_b, \\ \dot{y}_b &= v_b \sin \psi_b, \\ \dot{\psi}_b &= w_b, \end{aligned} \quad (3)$$

where v_b and w_b are the linear and angular velocities of TWR-B, respectively.

Suppose the TWR-A is aimed to track a trajectory of the TWR-B movement. This trajectory tracking is achieved by the TWR-A posture approaching the TWR-B posture at an instant time and converging as time goes to infinity. In this trajectory tracking, it is assumed that postures of both robots at any instant time are known and velocities of the TWR-B are given. Since the rate of change of the TWR-A posture is a function of the velocities, the trajectory tracking problem is therefore formulated to find proper velocities of the TWR-A.

Positions of both TWRs depicted in the Fig. 1 can be represented in a vector diagram as shown in Fig. 2. Their positions are thus expressed in terms of vectors r_a and r_b defined as follows:

$$r_a := \begin{bmatrix} x_a \\ y_a \end{bmatrix} \text{ and } r_b := \begin{bmatrix} x_b \\ y_b \end{bmatrix}. \quad (4)$$

Based on the vector diagram, the relationship of both vectors can then be written as

$$r_a + \tilde{r} = r_b \quad (5)$$

where $\tilde{r}$ denotes the position error of the TWR-A. This position error can be defined as follows:

$$\tilde{r} := r_b - r_a = \begin{bmatrix} x_b - x_a \\ y_b - y_a \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}. \quad (6)$$

Analogously, the orientation error of the TWR-A can be defined as follows:

$$\tilde{\psi} := \psi_b - \psi_a. \quad (7)$$

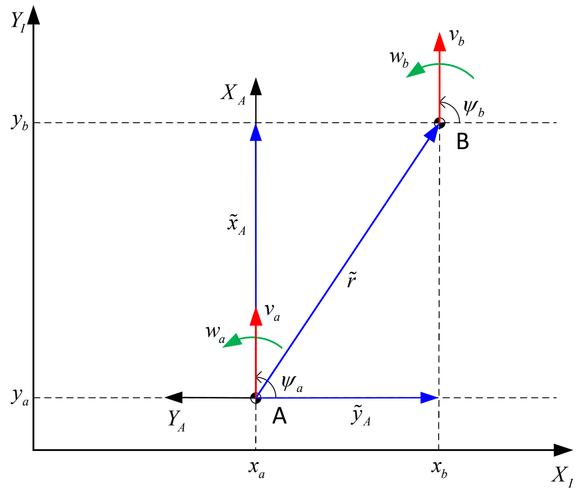

Fig. 3. A vector diagram of the TWR-A and TWR-B positions with respect to the TWR-A body coordinate system.

Thus, combining (6) and (7) results in the following vector:

$$\tilde{\xi} := \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\psi} \end{bmatrix}, \tag{8}$$

and call the $\tilde{\xi}$ as the posture error. This posture error represents a deviation of the TWR-A posture to the TWR-B posture. The TWR-A tracks the TWR-B trajectory if the posture error is minimized and vanished. The posture error is used as feedback information to calculate proper velocities of the TWR-A in order to tracks the TWR-B trajectory.

In addition to the inertial coordinate system, define a new coordinate system $X_AY_AZ_A$ and call it as the TWR-A body coordinate system. This coordinate system is sticking on the TWR-A and moves along the robot movements. The TWR-A body coordinate system is shown in Fig. 3, where the origin is located at the center of mass of TWR-A and the X_A axis is inline with the linear velocity v_a. The Y_A axis is perpendicular to the X_A axis such that the Z_A axis is perpendicular to the X_AY_A plane and pointing out of the Fig. 3.

The posture error $\tilde{\xi}$ in (8) was defined with respect to the inertial coordinate system. It can also be expressed in the TWR-A body coordinate system through a coordinate transformation. The transformation from inertial coordinate into the TWR-A body coordinate system is given by the following equation:

$$\tilde{\xi}_A = R_{AI}\tilde{\xi}. \tag{9}$$

The $\tilde{\xi}_A$ is the posture error represented in the TWR-A body coordinate system, R_{AI} is the transformation matrix form the inertial coordinate system into the TWR-A body coordinate system, and $\tilde{\xi}$ is the posture error represented in the inertial coordinate system. The transformation matrix R_{AI} is defined as follows [27]:

$$R_{AI} := \begin{bmatrix} \cos\psi_a & \sin\psi_a & 0 \\ -\sin\psi_a & \cos\psi_a & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Substituting (8) and (10) into (9) results in

$$\tilde{\xi}_A = \begin{bmatrix} \tilde{x}_A \\ \tilde{y}_A \\ \tilde{\psi}_A \end{bmatrix} = \begin{bmatrix} \cos\psi_a & \sin\psi_a & 0 \\ -\sin\psi_a & \cos\psi_a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\psi} \end{bmatrix}. \tag{11}$$

The expression of $\tilde{\xi}_A$ in (11) shows that the transformation does not change the orientation error,

$$\tilde{\psi}_A = \tilde{\psi}. \tag{12}$$

This is because the robot's rotational axes of both coordinate systems coincide. From the Figs. 1 to 3, we notice that the orientation angle of the TWR-A is 90°. Thus, applying this angle in to (11) results in:

$$\tilde{\xi}_A = \begin{bmatrix} \tilde{x}_A \\ \tilde{y}_A \\ \tilde{\psi}_A \end{bmatrix} = \begin{bmatrix} \tilde{y} \\ -\tilde{x} \\ \tilde{\psi} \end{bmatrix} \tag{13}$$

as shown in the vector diagram in Fig. 3.

Since the TWR-A is intended to track the TWR-B, the TWR-A has to move at certain velocities such that the posture error $\tilde{\xi}_A$ decreases and converges to zero as time goes to infinity. This can be achieved if the posture error dynamics is asymptotically stable. The posture error dynamics is formulated by differentiating $\tilde{\xi}_A$ with respect to time. It thus follows that a time derivative of (9) is given by

$$\dot{\tilde{\xi}}_A = \dot{R}_{AI}\tilde{\xi} + R_{AI}\dot{\tilde{\xi}}, \tag{14}$$

and the calculation result in

$$\begin{bmatrix} \dot{\tilde{x}}_A \\ \dot{\tilde{y}}_A \\ \dot{\tilde{\psi}}_A \end{bmatrix} = \begin{bmatrix} w_a\tilde{y}_A + v_b\cos\tilde{\psi}_A - v_a \\ -w_a\tilde{x}_A + v_b\sin\tilde{\psi}_A \\ w_b - w_a \end{bmatrix} \tag{15}$$

(see [27] for detailed calculation). To this end, the trajectory tracking problem is thus equivalent to finding the TWR-A velocities, v_a and w_a, such that the posture error dynamics (15) is asymptotically stable. Note that the posture error dynamics is manifestation of a nonlinear dynamical system. Thus, the trajectory tracking problem is not trivial to solve and its solution cannot be obtained in a straightforward manner.

The (15) can be presented in a general form of nonlinear dynamical system as follows:

$$\dot{z} = f(z, u) \tag{16}$$

where z represents the system state vector, u represents the system input vector, and $f(\cdot)$ represents a vector-valued nonlinear function. Here, z, u, and $f(\cdot)$ are given as follows:

$$z := \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \tilde{x}_A \\ \tilde{y}_A \\ \tilde{\psi}_A \end{bmatrix} \tag{17}$$

$$u := \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_a \\ w_a \end{bmatrix} \tag{18}$$

and

$$f(z, u) := \begin{bmatrix} u_2z_2 + v_b\cos z_3 - u_1 \\ -u_2z_1 + v_b\sin z_3 \\ w_b - u_2 \end{bmatrix}. \tag{19}$$

Suppose that $f(z,u)$ in (19) has an equilibrium point at the state z_q and the input u_q. Those are

$$z_q = \begin{bmatrix} z_{1_q} \\ z_{2_q} \\ z_{3_q} \end{bmatrix} \tag{20}$$

and

$$u_q = \begin{bmatrix} u_{1_q} \\ u_{2_q} \end{bmatrix}. \tag{21}$$

At the equilibrium point (z_q, u_q), we have

$$f(z_q, u_q) = 0 \tag{22}$$

such that the velocities of both robots satisfy

$$\begin{aligned} u_{2_q} z_{2_q} + v_b \cos z_{3_q} - u_{1_q} &= 0, & (23)\\ -u_{2_q} z_{1_q} + v_b \sin z_{3_q} &= 0, & (24)\\ w_b - u_{2_q} &= 0. & (25) \end{aligned}$$

Given the equilibrium point (z_q, u_q), one can express the nonlinear equation (19) in terms of the Taylor series [28] written as follows:

$$\begin{aligned} f(z,u) &= f(z_q,u_q) + \frac{\partial f(z_q,u_q)}{\partial z}(z - z_q) \\ &\quad + \frac{\partial f(z_q,u_q)}{\partial u}(u - u_q) + \text{H.O.T.} \end{aligned} \tag{26}$$

where H.O.T. stands for high-order terms. Thus, coefficients of the first-order terms in (26) are yielded as

$$\frac{\partial f(z_q,u_q)}{\partial z} = \begin{bmatrix} 0 & u_{2_q} & -v_b \sin z_{3_q} \\ -u_{2_q} & 0 & v_b \cos z_{3_q} \\ 0 & 0 & 0 \end{bmatrix} =: A, \tag{27}$$

$$\frac{\partial f(z_q,u_q)}{\partial u} = \begin{bmatrix} -1 & z_{2_q} \\ 0 & -z_{1_q} \\ 0 & -1 \end{bmatrix} =: B. \tag{28}$$

Let us now define deviations of the current system state z and the current system input u from the equilibrium point (z_q, u_q) respectively as the system state error $\tilde{z}$ and the system input error $\tilde{u}$:

$$\tilde{z} := z - z_q, \tag{29}$$

$$\tilde{u} := u - u_q. \tag{30}$$

Thus, neglecting the H.O.T. in (26) and taking the time derivative of the system state error $\tilde{z}$ in (29), one may obtain a linear dynamical equation as follows:

$$\dot{\tilde{z}} = A\tilde{z} + B\tilde{u}. \tag{31}$$

The dynamical equation (31) is a linear approximation of the the posture error dynamics in the neighbourhood of the equilibrium point (z_q, u_q) and is expressed in terms of a state-space model.

III. The Robot Trajectory Tracking Control Design

An admissible solution to the robot trajectory tracking problem requires asymptotic stability of the posture error dynamics at the equilibrium point (z_q, u_q). Through the linear approximation (31), the asymptotic stability is equivalently satisfied when the system matrix A is Hurwitz. However, this condition is not always fulfilled. Thus, in order to achieve the asymptotic stability, a state feedback controller can be synthesized to generate the control input $\tilde{u}$ such that the resulting closed-loop system matrix is Hurwitz. For this purpose, let us define the states feedback controller as follows:

$$\tilde{u} := -K\tilde{z} \tag{32}$$

where K is the controller gain matrix. Substituting (32) into (31) yields in

$$\dot{\tilde{z}} = (A - BK)\tilde{z} \tag{33}$$

which is the closed-loop system of (31). Therefore, it is necessary to design the state feedback controller (32) such that the closed-loop system (33) is asymptotically stable.

Various methods can be applied to design the state feedback controller (32) the system (31), and one of them is the linear quadratic regulator (LQR) method. Using the LQR method, one may synthesize the state feedback controller (32) by minimizing a quadratic cost function defined as follows:

$$J := \frac{1}{2}\int_0^\infty \left(\tilde{z}^T Q \tilde{z} + \tilde{u}^T R \tilde{u}\right) dt. \tag{34}$$

Here, $Q \geq 0$ is a positive semi-definite matrix and $R > 0$ is a positive definite matrix.

According to the optimal control theory (see e.g. [29]), minimization of the cost function J in (34) involves another function expressed as

$$H = \frac{1}{2}\left(\tilde{z}^T Q \tilde{z} + \tilde{u}^T R \tilde{u}\right) + \lambda^T \left(A\tilde{z} + B\tilde{u}\right), \tag{35}$$

which is referred to as the Hamiltonian function. Note that the parameter λ in (35) denotes a costate. When the the cost function J is minimized, these necessary conditions:

$$\frac{\partial H}{\partial \tilde{z}} = -\dot{\lambda} \quad \text{and} \quad \frac{\partial H}{\partial \tilde{u}} = 0 \tag{36}$$

are satisfied. It then follows that

$$\begin{aligned} \dot{\lambda} &= -\frac{\partial H}{\partial \tilde{z}} = -Q\tilde{z} - A^T \lambda & (37)\\ \tilde{u} &= -R^{-1} B^T \lambda. & (38) \end{aligned}$$

Now, substituting (38) into (31) yields

$$\dot{\tilde{z}} = A\tilde{z} - BR^{-1}B^T\lambda. \tag{39}$$

Furthermore, let us define the costate λ as

$$\lambda := P\tilde{z}, \tag{40}$$

where $P \geq 0$ is a positive semi-definite matrix. Then, substituting (40) into (37) and (39) results in

$$\dot{P}\tilde{z} + P\dot{\tilde{z}} = -Q\tilde{z} - A^T P \tilde{z}, \tag{41}$$

and

$$\dot{\tilde{z}} = A\tilde{z} - BR^{-1}B^T P\tilde{z}, \tag{42}$$

respectively. Finally, substituting (42) into (41) yields

$$(\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q)\tilde{z} = 0. \tag{43}$$

Since $\tilde{z}$ is not necessarily zero, the equation (43) must always be satisfied when

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0, \tag{44}$$

which is known as a Riccati differential equation. The matrix P is the only unknown in (44) and it is thus obtained as a solution to (44). Moreover, assuming a steady-state condition, where $\dot{P} = 0$, one is then able to recast the Riccati differential equation in (44) as an algebraic Riccati equation. That is,

$$0 = PA + A^T P - PBR^{-1}B^T P + Q. \tag{45}$$

Substituting the matrix P obtained from solving (45) into (40) and then (38) results in

$$\tilde{u} = -R^{-1}B^T P\tilde{z}. \tag{46}$$

This implies that the control input $\tilde{u}$ given in (46) is an optimal control input that asymptotically stabilizes the posture error dynamics (31). Therefore, referring to (32), one can straightforwardly determine that

$$K = R^{-1}B^T P \tag{47}$$

which is the optimal control gain matrix of the TTCS.

IV. Simulation Results

In this section, we demonstrate via numerical simulations what have been elaborated in Sections II and III. There are two TWRs in this simulations scenario and named as the TWR-A and the TWR-B. The designed TTCS is applied in the TWR-A, where the TWR-A is desired to track the TWR-B movements. The TWR-B is then called as the reference robot.

Let us assume that both robots have the same initial postures as follows:

$$\xi_a(0) = \xi_b(0) = \begin{bmatrix} 1 \\ 1 \\ 90^\circ \end{bmatrix} \tag{48}$$

and the initial velocities as follows:

$$\begin{aligned} v_a(0) &= v_b(0) = 1, && (49) \\ w_a(0) &= w_b(0) = 0. && (50) \end{aligned}$$

It is appropriate to consider this set of initial conditions as an equilibrium point of the posture error dynamics (19):

$$z_q = \begin{bmatrix} z_{1_q} \\ z_{2_q} \\ z_{3_q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{51}$$

$$u_q = \begin{bmatrix} u_{1_q} \\ u_{2_q} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{52}$$

Linearizing the posture error dynamics (19) at the equilibrium point (51) with the system input (52) results in the linearized posture error dynamic (31) with

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

Applying the LQR method as described in Section III, one is then able to design a trajectory tracking controller for the linearized system (31). In this case, the selected weighting-matrices are given as follows:

$$Q = \begin{bmatrix} 300 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 100 \end{bmatrix} \text{ and } R = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}. \tag{53}$$

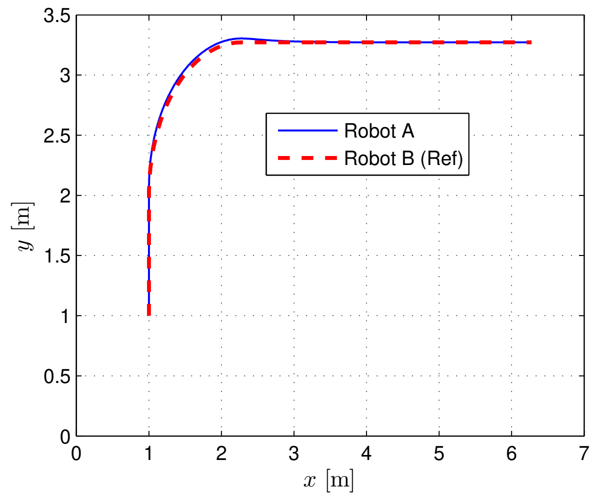

Fig. 4. The trajectory tracking performance of the TWR-A with respect to the TWR-B in the first simulation, where an initial posture of both TWRs are identical.

It thus follows that the controller gain matrix K of the state feedback controller (32) is obtained as

$$K = \begin{bmatrix} -5.4772 & 0 & 0 \\ 0 & -22.3607 & -12.0300 \end{bmatrix}. \tag{54}$$

Applying the state feedback controller to (31) yields a closed-loop trajectory tracking system with eigenvalues: $\lambda_1 = -5.4772$, $\lambda_2 = -9.7325$, and $\lambda_3 = -2.2975$. Since all of the eigenvalues are negative real numbers, the resulting closed-loop system is asymptotically stable. Therefore, applying the TTCS (32) with control gain (54) to the TWR-A will make the TWR-A's trajectory converge to the TWR-B's trajectory as a reference. This also implies that the posture error of the TWR-A with respect to the TWR-B's trajectory will decrease and eventually converge to zero.

Numerical simulations whose results are shown in Figs. 4 and 5 were performed to demonstrate trajectory tracking performance of the controller (32) using the control gain defined in (54). Fig. 4 shows that the TWR-A is able to track the TWR-B's trajectory. However, it is also noticed that although the trajectory tracking is precise when the TWR-A follows a straight line path, a small tracking error appears when the TWR-A performs a turning motion. The tracking error consists of the position errors and the orientation error as shown in the Fig. 5. Here, the position error x_e in the x direction has an amplitude of approximately 0.025 m, but it vanishes within 2.5 s; and the position error y_e in the y direction has an amplitude of approximately 0.034 m and vanishes within 4.0 s. Magnitudes of those position errors are relatively small as compared to the turning radius 1.27 m. Moreover, the orientation error ψ_e has an amplitude of approximately 3.0° and vanishes within 5.0 s. The results depicted in Fig. 5 thus confirm the asymptotic stability of the closed-loop TTCS.

While the previous batch of simulation results was generated by setting the initial postures of both robots to be identical, the second batch of simulation results is presented to evaluate the performance of the TTCS when the initial

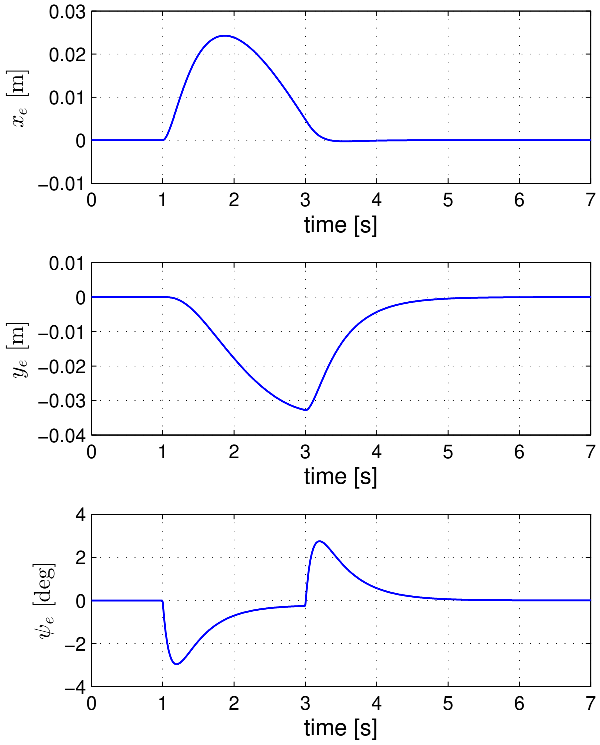

Fig. 5. The posture error of the TWR-A with respect to the TWR-B's posture in the fist simulation, where both robots have the same initial postures.

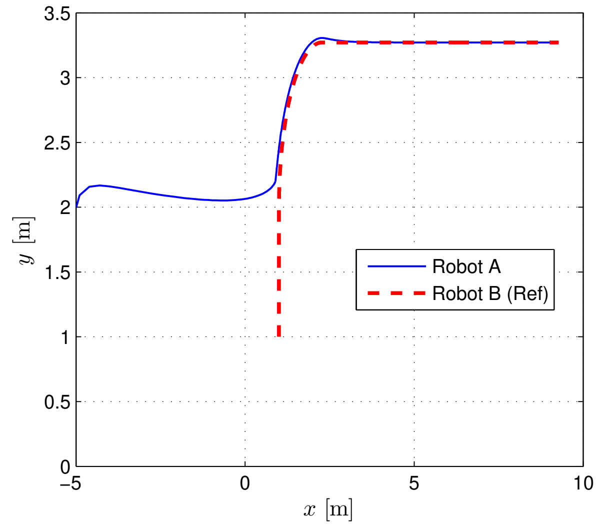

Fig. 6. The trajectory tracking performance of the TWR-A with respect to the TWR-B in the second simulation, where the initial postures of both TWRs are different.

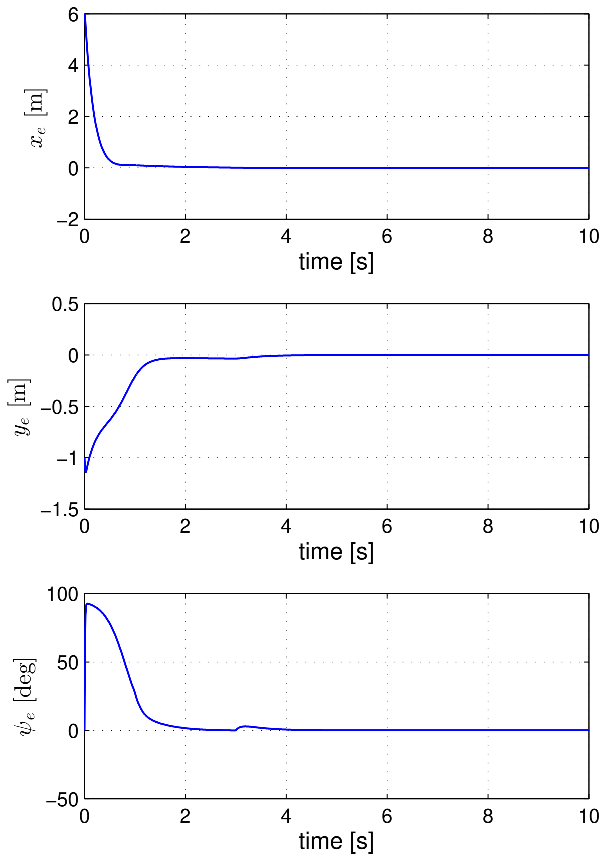

Fig. 7. The posture error of the TWR-A with respect to the TWR-B's posture in the second simulation, where both robots have different initial postures.

postures of both robots are not identical. Those are,

$$\xi_a(0) = \begin{bmatrix} -5 \\ 2 \\ 90^\circ \end{bmatrix} \text{ and } \xi_b(0) = \begin{bmatrix} 1 \\ 1 \\ 90^\circ \end{bmatrix}.$$

Nonetheless, the initial velocities of both robots were set to be identical as follows:

$$\begin{aligned} v_a(0) &= v_b(0) = 1, \\ w_a(0) &= w_b(0) = 0. \end{aligned}$$

The results of the second batch of simulation are shown in Figs. 6 and 7. Here, Fig. 6 shows that the TWR-A was able to track the TWR-B's trajectory although the initial position of the TWR-A is relatively far from that of the TWR-B. It is also shown in Fig. 7 that the posture error of TWR-A with respect to that of the TWR-B decreases and converges to zero as time goes infinity. Note that the posture error converges to zero relatively fast because no constraint was imposed on the robot's velocities.

V. Conclusions

A new approach to the kinematic modeling of the TWR lateral motion based on the vector diagram has been presented in this paper. Through this modeling, we show how to transform the robot trajectory tracking problem into the stabilization problem of the posture error dynamics. This signifies the merit of the kinematic modeling as it then enables one to synthesize the state feedback controller for stabilizing the posture error dynamics. It has been demonstrated that the state feedback controller can be designed via the LQR method based on the linearized posture error dynamics. The state feedback controller plays such an important role in the TTCS that the TWR can track a reference trajectory. Performance of the resulting TTCS has been demonstrated via numerical simulations. The simulation results show that despite some errors, the robot equipped with the TTCS is capable of tracking the reference trajectory regardless of the initial postures of the robots involved.

REFERENCES

[1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. in International Conference on Robotics and Automation*. IEEE, 1990, pp. 384–389.

[2] Y. S. Ha and S. Yuta, "Trajectory tracking control for navigation of self-contained mobile inverse pendulum," in *Proc. in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 3. IEEE, 1994, pp. 1875–1882.

[3] Y.-S. Ha *et al.*, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robotics and Autonomous Systems*, vol. 17, no. 1, pp. 65–80, 1996.

[4] T. Zhang, Q. Li, C. S. Zhang, H. W. Liang, P. Li, T. M. Wang, S. Li, Y. L. Zhu, and C. Wu, "Current trends in the development of intelligent unmanned autonomous systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 68–85, 2017.

[5] C. K. On and J. Teo, "Artificial neural controller synthesis in autonomous mobile cognition," *IAENG International Journal of Computer Science*, vol. 36, no. 4, pp. 240–252, 2009.

[6] A. Gorbenko and V. Popov, "Visual landmark selection for mobile robot navigation," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 134–142, 2013.

[7] Y. Chen, R. Huang, and Y. Zhu, "A cumulative error suppression method for UAV visual positioning system based on historical visiting information." *Engineering Letters*, vol. 25, no. 4, pp. 424–430, 2017.

[8] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*, vol. 37, no. 1, pp. 89–103, 2013.

[9] N. Uddin, "Lyapunov-based control system design of two-wheeled robot," in *Proc. in Computer, Control, Informatics and Its Applications (IC3INA)*. IEEE, 2017, pp. 121–125.

[10] ——, "Adaptive control system design for two-wheeled robot stabilization," in *Proc. in 12th South East Asian Technical University Consortium (SEATUC)*, vol. 1. IEEE, 2018, pp. 1–5.

[11] E. Koyanagi, S. Lida, and S. Yuta, "A wheeled inverse pendulum type self-contained mobile robot and its two-dimensional trajectory control," *Proc. in 2nd International Symposium on Measurement and Control in Robotics (ISMCR)*, pp. 891–898, 1992.

[12] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, 2005.

[13] Z. Li and Y. Zhang, "Robust adaptive motion/force control for wheeled inverted pendulums," *Automatica*, vol. 46, no. 8, pp. 1346–1353, 2010.

[14] N. Uddin, "A two-wheeled robot trajectory tracking control system design based on poles domination approach." *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp. 154–161, 2020.

[15] R. Cui, J. Guo, and Z. Mao, "Adaptive backstepping control of wheeled inverted pendulums models," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 501–511, 2015.

[16] K. Kankhunthodl, V. Kongratana, A. Numsomran, and V. Tipsuwanporn, "Self-balancing robot control using fractional-order PID controller," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019, IMECS 2019*, 2019, pp. 77–82.

[17] S. Kim and S. Kwon, "Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot," *International Journal of Control, Automation and Systems*, vol. 13, no. 4, pp. 926–933, 2015.

[18] F. Grasser, A. D'arrigo, S. Colombi, and A. C. Rufer, "Joe: a mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–114, 2002.

[19] S. Miao and Q. Cao, "Modeling of self-tilt-up motion for a two-wheeled inverted pendulum," *Industrial Robot: An International Journal*, vol. 38, no. 1, pp. 76–85, 2011.

[20] D. Choi and J.-H. Oh, "Human-friendly motion control of a wheeled inverted pendulum by reduced-order disturbance observer," in *Proc. in International Conference on Robotics and Automation*. IEEE, 2008, pp. 2521–2526.

[21] P. Petrov and M. Parent, "Dynamic modeling and adaptive motion control of a two-wheeled self-balancing vehicle for personal transport," in *Proc. in 13th International Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1013–1018.

[22] S. Jeong and T. Takahashi, "Wheeled inverted pendulum type assistant robot: design concept and mobile control," *Intelligent Service Robotics*, vol. 1, no. 4, pp. 313–320, 2008.

[23] A. Salerno and J. Angeles, "A new family of two-wheeled mobile robots: Modeling and controllability," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 169–173, 2007.

[24] H. Azizan, M. Jafarinasab, S. Behbahani, and M. Danesh, "Fuzzy control based on lmi approach and fuzzy interpretation of the rider input for two wheeled balancing human transporter," in *Proc. in 8th IEEE International Conference on Control and Automation*. IEEE, 2010, pp. 192–197.

[25] Y. Kim, S. H. Kim, and Y. K. Kwak, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *Journal of Intelligent and Robotic Systems*, vol. 44, no. 1, pp. 25–46, 2005.

[26] A. Sikander and R. Prasad, "Reduced order modelling based control of two wheeled mobile robot," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1057–1067, 2019.

[27] N. Uddin, "Trajectory tracking control system design for autonomous two-wheeled robot," *Jurnal Infotel*, vol. 10, no. 3, pp. 90–97, 2018.

[28] G. B. Thomas, M. D. Weir, J. Hass, C. Heil, and A. Behn, *Thomas' Calculus: Early Transcendentals*. Pearson, 2016.

[29] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. New Jersey, USA: John Wiley & Sons, 2012.