sensors

Article

A Sensor Fusion Based Nonholonomic Wheeled Mobile Robot for Tracking Control

Shun-Hung Tsai [1], Li-Hsiang Kao [1], Hung-Yi Lin [2], Ta-Chun Lin [1], Yu-Lin Song [3,*] and Luh-Maan Chang [2]

[1] Graduate Institute Automation Technology, National Taipei University of Technology, Taipei 10608, Taiwan; shtsai@ntut.edu.tw (S.-H.T.); t109669011@ntut.edu.tw (L.-H.K.); eugenesong1130@gmail.com (T.-C.L.)

[2] High-Tech Facility Research Center, Department of Civil Engineering, National Taiwan University, Zhubei 30264, Taiwan; hungyilin@ntu.edu.tw (H.-Y.L.); luhchang@ntu.edu.tw (L.-M.C.)

[3] Department of Bioinformatics and Medical Engineering, Asia University, Taichung 41354, Taiwan

* Correspondence: ylsong@asia.edu.tw or d87222007@ntu.edu.tw; Tel.: +886-3-668-1274

Received: 30 October 2020; Accepted: 4 December 2020; Published: 9 December 2020

Abstract: In this paper, a detail design procedure of the real-time trajectory tracking for the nonholonomic wheeled mobile robot (NWMR) is proposed. A 9-axis micro electro-mechanical systems (MEMS) inertial measurement unit (IMU) sensor is used to measure the posture of the NWMR, the position information of NWMR and the hand-held device are acquired by global positioning system (GPS) and then transmit via radio frequency (RF) module. In addition, in order to avoid the gimbal lock produced by the posture computation from Euler angles, the quaternion is utilized to compute the posture of the NWMR. Furthermore, the Kalman filter is used to filter out the readout noise of the GPS and calculate the position of NWMR and then track the object. The simulation results show the posture error between the NWMR and the hand-held device can converge to zero after 3.928 seconds for the dynamic tracking. Lastly, the experimental results show the validation and feasibility of the proposed results.

Keywords: nonholonomic; wheeled mobile robot (WMR); tracking control; global positioning system (GPS); radio frequency (RF); Kalman filter; quaternion; trajectory tracking

1. Introduction

Wheeled mobile robot (WMR) has been utilized in many fields such as the automated guided vehicle (AGV) [1,2], robotic cleaner [3], exploration robot [4,5] recently. The WMR has the capability to respond flexibility by itself in various complex unknown environments. 9-axis IMU sensor integrated with a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer is used to estimate linear velocity, angular velocity and orientation of the body relative to the global reference frame. Hence, the posture of the body can be obtained [6–8]. The free, open, and dependable nature of GPS has led to the development of applications affecting every aspect of modern life. GPS is capable of receiving signal from GPS satellites and someone can use these useful received information to calculate the geographical position of the body [9,10].

Different types of the WMR had been proposed. Omnidirectional WMR (OWMR) is one of the WMR that has the advantage of multimode action and high mobility [11–15]. NWMR is another type of the WMR. Owing to the nonholonomic constraint, the moving characteristic of the NWMR is rolling but not sliding, as a result, NWMR can not move laterally and difficult to turn the body in any direction. The goal of the trajectory tracking is to design a motion controller of WMR for adjusting linear velocity and angular velocity, so the WMR can track to the desired trajectory. The aim of the tracking controller is to minimize the tracking errors between the real trajectory and the desired trajectory. These errors are arisen from slippage, disturbances and the measured error of sensor readout. A cross-coupled controller is presented in [16]. The controller is used to reduce wheel slip of the vehicle. Sun and Yan presented an evaluation method to estimate the unknown disturbance. The estimated information is compensated by the controller of the NWMR, hence, the tracking error of the NWMR is repressed [17,18]. In [19], Rayguru proposed a robust-observer based sliding mode controller to achieve the motion control task in the presence of incomplete state measurements and sensor inaccuracies. A two-layer lateral path tracking controller are presented in [20], the upper-layer controller is implemented with a linear time-varying model predictive control (LTV-MPC) algorithm, the lower-layer controller is implemented by a radial basis function neural network proportion-integral-derivative (RBFNN-PID) algorithm. The advantage of the two-layer controller is that controller can track the reference paths accurately while ensuring the stability of the vehicle, however, this type of the tracking controller suffers from high computation. In [21], a single controller for simultaneous stabilization and trajectory tracking of NWMR is proposed. A controller with time-varying parameters is designed to stabilize the error system. Then, the geometric analysis method is used to guarantee that the controller inputs stay in the restricted input domain. Hence the controller can prevent the actuator from saturation and the stability of the system is improved. Kanayama and Sanhoury presented the design of the tracking controllers with stability [22,23]. These two types of tracking controller based on the direct Lyapunov method are used for solving the lateral error issue. The advantages of these two tracking controllers are less computational complexity and fast response to the tracking error.

The contribution of this paper is to implement the real-time trajectory tracking for NWMR. IMU, GPS and RF sensors are integrated into NWMR. In order to be implemented into the resource limited embedded system, a low computational complexity, fast response to tracking error and stability of tracking controller were key factors and need be traded off. Hence, the conventional Kalman filter [24] and tracking controller proposed by [23] are utilized. The rest of this paper is organized as follows: Section 2 illustrates the system function block and system flow chart of proposed NWMR. Section 3 presents the kinematic model of the NWMR, the design of the tracking controller on the NWMR is discussed. In Section 4, the algorithms for the sensors equipped on the NWMR are introduced. Section 5 demonstrates the simulations and experimental results of the developed NWMR. Finally, conclusions and future works are given in Section 6.

2. The Proposed Architecture

In this section, the proposed NWMR is described in Section 2.1. Section 2.2 illustrates the system flow chart of the NWMR.

2.1. The Block Diagram

The block diagram of the proposed NWMR system is shown in Figure 1. In the NWMR, a 9-axis sensor is utilized to acquire the current posture of the robot. Two GPS modules are used to obtain the current positions of the NWMR and the target hand-held device, respectively. RF transmitter in the hand-held device transmits the position information of the tracking object to the NWMR. The tracking controller and sensor fusion algorithms are implemented in the STM32F429 microprocessor [25].

2.2. The Flow Charts of Trajectory Tracking

Figure 2 shows the system flow chart of the NWNR. The current position data of NWNR is obtained by GPS and is processed by the Kalman filter. The current posture data of NWNR is acquired by the 9-axis IMU sensor and then is processed by the Madgwick's data fusion algorithm [26,27]. RF module receive the current position data from target hand-held device. Using these three data as input parameters of the tracking controller, the tracking controller calculates the shortest distance from NWNR to the hand-held device. While the distance between the NWNR and the hand-held device is smaller than 1.5 m, the tracking controller will stop the tracking task, and the mission of trajectory tracking is finished.

Figure 3 shows the system flow chart of the hand-held device which is considered as the tracking object. The position data of hand-held device is obtained by GPS and is refined by the the Kalman filter. After processing, the data is sent out by the RF module.

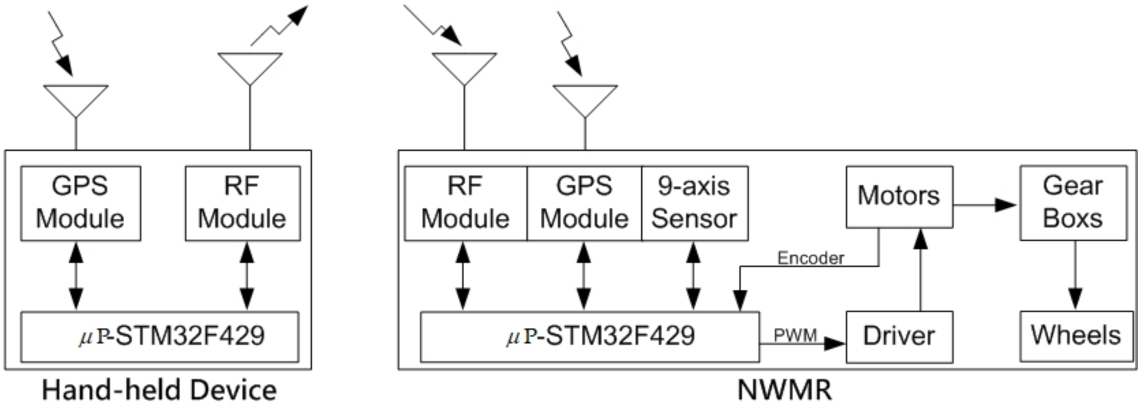


Figure 1. The block diagram of the proposed NWMR system.

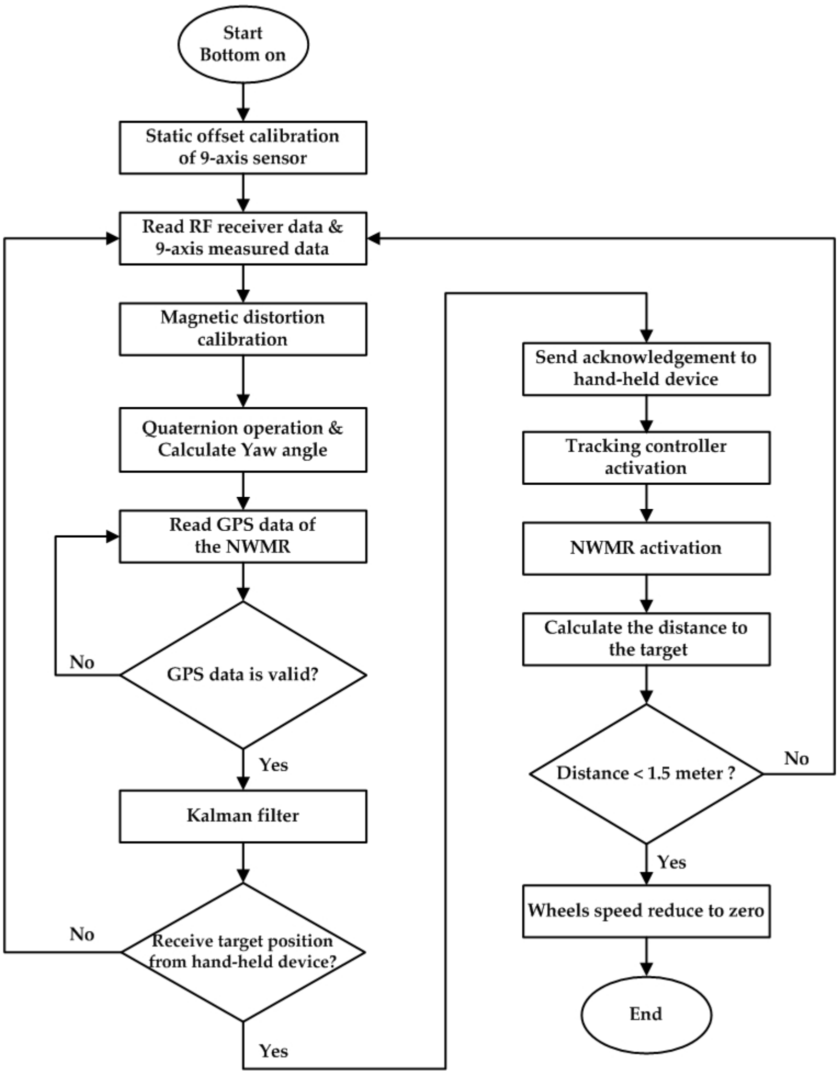

Start Bottom on
Static offset calibration of 9-axis sensor
Read RF receiver data & 9-axis measured data
Magnetic distortion calibration
Quaternion operation & Calculate Yaw angle
Read GPS data of the NWMR
GPS data is valid?
No
Yes
Kalman filter
Receive target position from hand-held device?
No
Yes
Send acknowledgement to hand-held device
Tracking controller activation
NWMR activation
Calculate the distance to the target
Distance < 1.5 meter ?
No
Yes
Wheels speed reduce to zero
End

Figure 2. The flow chart of the trajectory tracking for the NWMR.

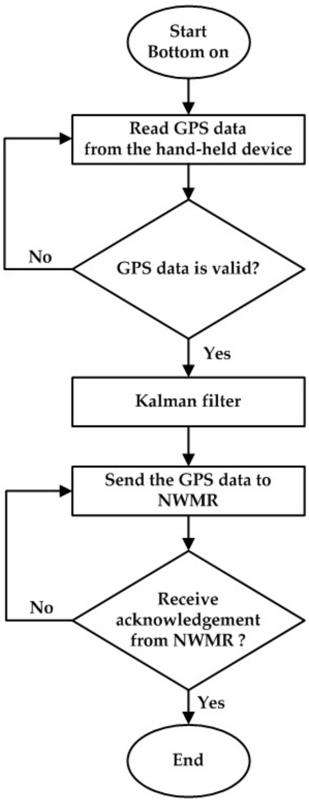


Figure 3. The flow chart of the trajectory tracking for the hand-held device.

3. Kinematic Model and Tracking Controller Design

As depicted in Figure 4, the NWMR considered in this paper is a kind of four-wheeled robot. This robot body has symmetric shape and the center of mass is at the geometric center C of the body. Two rear wheels are driven differentially by motors, two front wheels prevent the NWMR from tipping over while the robot moves on a plane. In this paper, it is assumed that the motion of two front wheels can be ignored in dynamics of the NWMR.

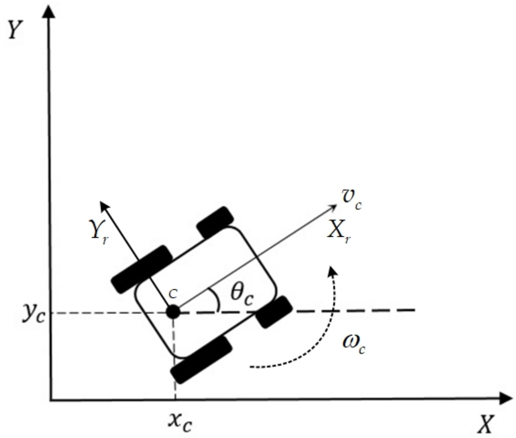


Figure 4. The posture of the NWMR.

3.1. *Kinematic Model of the NWMR and Tracking Problem Representation*

Figure 4 shows the posture of the NWMR in the global X-Y coordinate. The local coordinate system is fixed to the NWMR with point C as the origin. (x_c, y_c) is the current position of the geometric center C in the global X-Y coordinate, θ_c is the angle between the X-axis and X_r -axis, it is represented as the heading direction of the NWMR, v_c denotes the linear velocity of the NWMR in the direction of X_r -axis and ω_c is the angular velocity of the robot.

Consider the kinematic model of the NWMR, the posture variable of NWMR, r_c , is defined as

$$r_c = [x_c \;\; y_c \;\; \theta_c]^T \tag{1}$$

The moving of the robot is controlled by v_c and ω_c, the input state variable of the robot, u_c , is therefore defined as

$$u_c = [v_c \;\; \omega_c]^T \tag{2}$$

Under the nonholonomic constraint,

$$\dot{x}_c \sin\theta_c - \dot{y}_c \cos\theta_c = 0 \tag{3}$$

The kinematic model of the NWMR can be expressed by [28]

$$\dot{r}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix} u_c = \begin{bmatrix} v_c \cos\theta_c \\ v_c \sin\theta_c \\ \omega_c \end{bmatrix} \tag{4}$$

The position information of the tracking object is obtained by the hand-held device via GPS module. As shown in Figure 5, it is assumed that the hand-held device is circular shape and the center of mass is located at the geometric center D of the device.

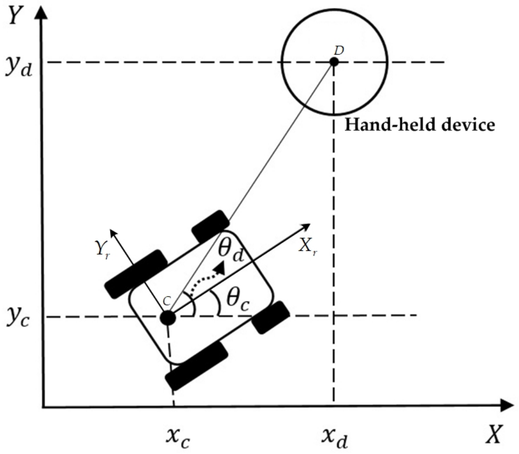


Figure 5. The posture of the hand-held device.

$(x_d\ , y_d)$ is the current position of the geometric center D in the X-Y coordinate. θ_d is the angle between the X-axis and the straight line that pass through point C and point D, which is defined as

$$\theta_d = \frac{180}{\pi} tan^{-1}(\frac{|y_d - y_c|}{|x_d - x_c|}) \tag{5}$$

The posture of the hand-held device, r_d , is defined as

$$r_d = [x_d \;\; y_d \;\; \theta_d]^T \tag{6}$$

For simplicity, it is assumed that the nonholonomic constraint of the hand-held device can be written as

$$\dot{x}_d \sin\theta_d - \dot{y}_d \cos\theta_d = 0 \tag{7}$$

Two postures are used in this tracking control system; One is current posture, P_c , and the other is reference posture, P_d . Current posture is the real posture of the NWMR, reference posture is the target posture, i.e., the real posture of the hand-held device, hence

$$P_c = r_c \ \ ; \ \ P_d = r_d \tag{8}$$

As illustrated in Figure 6, the posture error, P_e , between the reference posture and the current posture in the local $X_r - Y_r$ coordinate is expressed as

$$P_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} (P_d - P_c) = R(\theta)(P_d - P_c) \tag{9}$$

where x_e is the error in the X_r direction of the robot, y_e is the error in the Y_r direction of the robot, θ_e is the orientation error, i.e., $\theta_e = \theta_d - \theta_c$, R($\theta$) is rotation matrix and it is expressed as the orientation of the current posture, P_c , with respect to the reference posture, P_d.

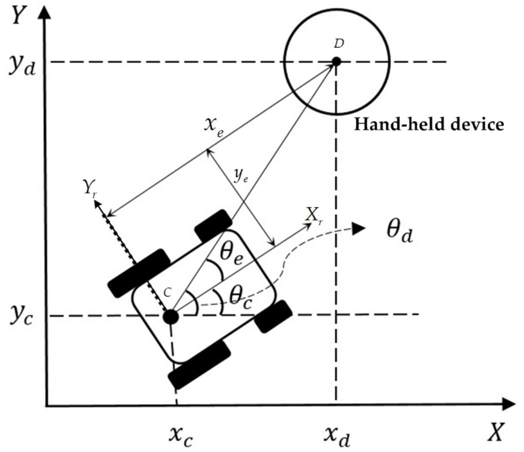


Figure 6. The posture error of the NWMR and the hand-held device.

By differentiating (9), we have

$$\begin{aligned} \dot{x}_e &= (\dot{x}_d - \dot{x}_c)\cos\theta_c + (\dot{y}_d - \dot{y}_c)\sin\theta_c - (x_d - x_c)\dot{\theta}_c \sin\theta_c + (y_d - y_c)\dot{\theta}_c \cos\theta_c \\ &= \dot{x}_d \cos\theta_c + \dot{y}_d \sin\theta_c - (v_c \cos^2\theta_c + v_c \sin^2\theta_c) + y_e\omega_c \\ &= \dot{x}_d \cos\theta_c + \dot{y}_d \sin\theta_c - v_c + y_e\omega_c \\ &= \dot{x}_d \cos(\theta_d - \theta_e) + \dot{y}_d \sin(\theta_d - \theta_e) - v_c + y_e\omega_c \end{aligned} \tag{10}$$

Substituting (7) and the kinematic model of the NWMR at point D into (10), we obtain

$$\dot{x}_e = v_d \cos\theta_e - v_c + y_e\omega_c \tag{11}$$

Similarly,

$$\begin{aligned} \dot{y}_e &= -(\dot{x}_d - \dot{x}_c)\sin\theta_c + (\dot{y}_d - \dot{y}_c)\cos\theta_c - (x_d - x_c)\dot{\theta}_c \cos\theta_c - (y_d - y_c)\dot{\theta}_c \sin\theta_c \\ &= -\dot{x}_d \sin\theta_c + \dot{y}_d \cos\theta_c - x_e\omega_c \\ &= v_d \sin\theta_e - x_e\omega_c \end{aligned} \tag{12}$$

$$\dot{\theta}_e = \dot{\theta}_d - \dot{\theta}_c = \omega_d - \omega_c \tag{13}$$

Combining (11) to (13), the differential equation of the posture error of NWMR with respect to the hand-held device is obtained

$$\dot{P}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_d \cos\theta_e - v_c + y_e\omega_c \\ v_d \sin\theta_e - x_e\omega_c \\ \omega_d - \omega_c \end{bmatrix} \tag{14}$$

3.2. The Tracking Controller Design

The function block of the tracking controller of the NWMR is demonstrated in Figure 7. Firstly, the posture error $P_e(t)$ between $P_d(t)$ and $P_c(t)$ at time t is obtained by rotation matrix from (9). Based on the variable $P_e(t)$, the tracking controller is designed to stabilize the kinematic model of NWMR in concurrent with unknown error or disturbances N_{noise}. The variable $u_d(t) = [v_d(t) \;\; \omega_d(t)]^T$ are the linear velocity and the angular velocity of the reference posture and is acted as the input of the tracking controller. Finally, the updated posture $P_c(t)$ of the NWMR generated by an integrator is used for next posture error correction.

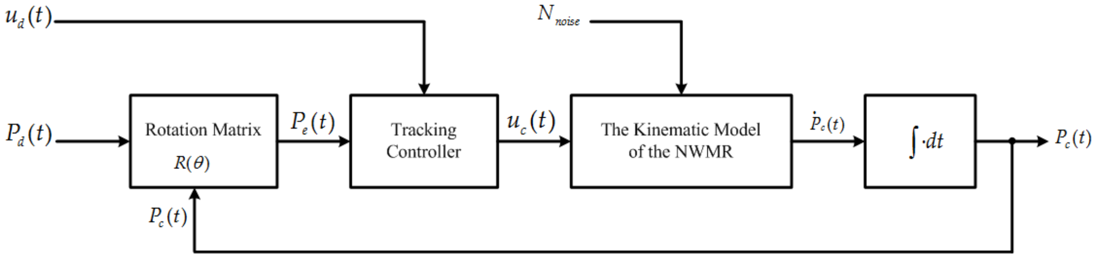


Figure 7. The function block of the tracking controller design.

The tracking controller proposed by [23] is used for the developed NWMR, we have

$$u_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_d \cos\theta_e + k_1 x_e + k_4 sign(x_e) y_e^2 \\ \omega_d + v_d(k_2 y_e + k_3 \sin\theta_e) \end{bmatrix} \tag{15}$$

where k_1, k_2, k_3 and k_4 are adjustable gain coefficients, $sign(x_e)$ is described as:

$$sign(x_e) = \begin{cases} -1, \; x_e < 0 \\ \;\;1, \; x_e \geq 0 \end{cases} \tag{16}$$

By using Lyapunov function and Routh-Hurwitz Criterion to examine the stability of the tracking controller, the posture error $P_e = 0$ is uniformly asymptotical stable over interval $[0, \infty)$ under the conditions: (a) v_d and ω_d are continuous, (b) v_d, ω_d, k_1, k_2, k_3 and k_4 are all bounded and (c) $\dot{v}_d$ and $\dot{\omega}_d$ are sufficiently small [22] .

4. Signal Processing Algorithms

In this section, the sensor fusion algorithms for the NWMR and hand-held device are introduced.

4.1. Madgwick's Data Fusion Algorithm for the 9-Axis IMU Sensor

The accurate measurement of orientation plays a key factor in the design of the tracking control of the NWMR. A MEMS gyroscope outputs angular velocity and the data is used to compute the orientation of the robot. However, this sensor suffers from an accumulated error. An accelerometer and magnetometer measure the earth's gravity and magnetic field, respectively. These two sensors provide an absolute reference of orientation. Nevertheless, these sensors subject to high levels of noise.

The goal of the data fusion algorithm is to estimate a single orientation through the optimal data fusion of gyroscope, accelerometer and magnetometer measurements.

Euler angles are three angles and can be used to describe the orientation of a mobile frame of reference. However, in some motion case, Euler angles suffer from gimbal lock mechanism [29]. Quaternions developed by Hamilton is a mathematical number system that can be treated as the other representation of the three-dimensional space [30]. Quaternion method is well used in mechanics, computer graphics, crystallographic texture analysis in three-dimensional space for preventing from the gimbal lock mechanism of the Euler angles. In inertial navigation system (IMS), it uses quaternion methods to derive the orientation and the velocity of vehicle. In this paper, we also use quaternion method to calculate three-dimensional motion of the NWMR.

The data fusion algorithm proposed by Madgwick is applied for the 9-axis [26,27]. Figure 8 shows the block diagram of the Madgwick's fusion algorithm. The measured output data of the 3-axis sensor ${}^{S}\hat{d}$ in the sensor frame can be transfer to the earth frame ${}^{E}\hat{d}$ by using a quaternion ${}^{S}_{E}\hat{q}$, yields

$$\begin{aligned} &{}^{E}\hat{d} = {}^{S}_{E}\hat{q} \otimes {}^{S}\hat{d} \otimes {}^{S}_{E}\hat{q}^{*} \\ &{}^{E}\hat{d} = [0\ d_x\ d_y\ d_z] \\ &{}^{S}_{E}\hat{q} = [q_1\ \ q_2\ q_3\ \ q_4] \\ &{}^{E}_{S}\hat{q}^{*} = {}^{S}_{E}\hat{q} = [q_1\ (-q_2)\ (-q_3)\ (-q_4)] \end{aligned} \tag{17}$$

where d_x , d_y and d_z are the three dimensional vector components of ${}^{E}\hat{d}$, which aligns a predefined reference direction in the earth frame, $\otimes$ is quaternion product, $q_1\ \ q_2\ \ q_3$ and q_4 are the components of ${}^{S}_{E}\hat{q}$, and ${}^{S}_{E}\hat{q}^{*}$ is conjugate of ${}^{S}_{E}\hat{q}$. A 3-axis gyroscope measures the angular velocity of robot, the angular rate ${}^{S}\omega$ is define as

$${}^{S}\omega = [0\ \ {}^{S}\omega_x\ \ {}^{S}\omega_y\ \ {}^{S}\omega_z] \tag{18}$$

where ${}^{S}\omega_x$, ${}^{S}\omega_y$ and ${}^{S}\omega_z$ is the angular velocity of the X, Y, Z axes in the sensor frame, respectively. Derivative of the quaternion at time t describes the rate of the change of orientation, ${}^{S}_{E}\dot{\hat{q}}_{\omega\_est}(t)$, and can be calculated as

$${}^{S}_{E}\dot{\hat{q}}_{\omega\_est}(t) = \frac{1}{2}{}^{S}_{E}\hat{q}_{est}(t-1) \otimes {}^{S}\omega(t) \tag{19}$$

where ${}^{S}_{E}\hat{q}_{est}(t-1)$ is prior estimate of orientation, the sub-script ω indicates that the quaternion is calculated from gyroscope and ${}^{S}\omega(t)$ is angular velocity measured at time t. If the angular velocity of the sensor is not constant, the derivative of quaternion in (19) can be used to estimate the current orientation of the robot, which is shown as

$${}^{S}_{E}\hat{q}_{\omega\_est}(t) = {}^{S}_{E}\hat{q}_{est}(t-1) + {}^{S}_{E}\dot{\hat{q}}_{\omega\_est}(t)\Delta t \tag{20}$$

where Δt is sampling period. Substituting (19) into (20), yields

$${}^{S}_{E}\hat{q}_{\omega\_est}(t) = {}^{S}_{E}\hat{q}_{est}(t-1) + \frac{1}{2}{}^{S}_{E}\hat{q}_{est}(t-1) \otimes {}^{S}\omega(t)\Delta t \tag{21}$$

Contrary to the orientation calculated from the angular velocity, gravity and magnetic field can also be applied to estimate orientation. Using (17) and quaternion operations, the measured direction vector in the earth frame can also be rotated to the sensor frame, therefore

$${}^{S}\hat{d} = {}^{S}_{E}\hat{q}^{*} \otimes {}^{E}\hat{d} \otimes {}^{S}_{E}\hat{q} \tag{22}$$

In order to estimate the quaternion, the difference between a known vector in the sensor frame and a measured vector in sensor frame is minimized. In other words,

$$min\ f({}^{S}\hat{d}_{est}, {}^{S}\hat{S}) = min\ f({}^{S}_{E}\hat{q}^{*}_{est} \otimes {}^{E}\hat{d} \otimes {}^{S}_{E}\hat{q}_{est} - {}^{S}\hat{S}) = min\ f({}^{S}_{E}\hat{q}_{est}, {}^{E}\hat{d}, {}^{S}\hat{S}) \tag{23}$$

where ${}^{S}\hat{d}_{est}$ is a sensor frame's estimated vector, ${}^{S}_{E}\hat{q}_{est}$ is the estimated quaternion and ${}^{S}\hat{S}$ is measured vector in sensor frame. Madgwick uses the gradient descent algorithm to iterate the estimated quaternion ${}^{S}_{E}\hat{q}_{est}(t)$ based on an initial guess quaternion ${}^{S}_{E}\hat{q}_{est}(0)$ and a step-size μ [26,27], the ${}^{S}_{E}\hat{q}_{est}(t)$ can be expressed as

$$ {}^{S}_{E}\hat{q}_{est}(t) = {}^{S}_{E}\hat{q}_{est}(t-1) - \mu \frac{\nabla f({}^{S}\hat{d}_{est}(t), {}^{S}\hat{S}(t))}{\|\nabla f({}^{S}\hat{d}_{est}(t), {}^{S}\hat{S}(t))\|} \tag{24} $$

$$ \nabla f({}^{S}\hat{d}_{est}(t), {}^{S}\hat{S}(t)) = J^{T}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}(t)) f({}^{S}_{E}\hat{q}_{est}(t-1){}^{E}\hat{d}(t), {}^{S}\hat{S}(t)) \tag{25} $$

where (25) computes the gradient of the solution surface defined by objective function $f(\cdot)$ and its Jacobian $J(\cdot)$.

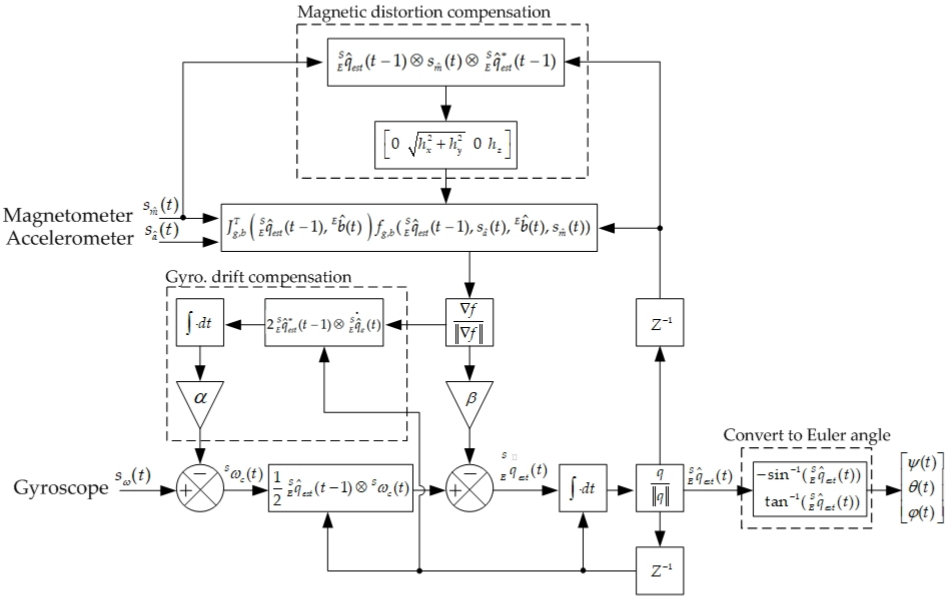

Magnetic distortion compensation
${}^{S}_{E}\hat{q}_{est}(t-1) \otimes s_{\hat{m}}(t) \otimes {}^{S}_{E}\hat{q}^{*}_{est}(t-1)$
$[0 \ \sqrt{h_x^2+h_y^2} \ 0 \ h_z]$
Magnetometer
Accelerometer
$s_{\hat{m}}(t)$
$s_{\hat{a}}(t)$
$J^{T}_{g,b}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{b}(t)) f_{g,b}({}^{S}_{E}\hat{q}_{est}(t-1), s_{\hat{a}}(t), {}^{E}\hat{b}(t), s_{\hat{m}}(t))$
Gyro. drift compensation
$\int \cdot dt$
$2\,{}^{S}_{E}\hat{q}^{*}_{est}(t-1) \otimes {}^{S}_{E}\dot{\hat{q}}_{\varepsilon}(t)$
$\frac{\nabla f}{\|\nabla f\|}$
Z^{-1}
α
β
Convert to Euler angle
Gyroscope
$s_{\omega}(t)$
${}^{S}\omega_{c}(t)$
$\frac{1}{2}\,{}^{S}_{E}\hat{q}_{est}(t-1) \otimes {}^{S}\omega_{c}(t)$
${}^{S}_{E}q_{est}(t)$
$\int \cdot dt$
$\frac{q}{\|q\|}$
${}^{S}_{E}\hat{q}_{est}(t)$
$-\sin^{-1}({}^{S}_{E}\hat{q}_{est}(t))$
$\tan^{-1}({}^{S}_{E}\hat{q}_{est}(t))$
$\psi(t)$
$\theta(t)$
$\varphi(t)$
Z^{-1}

Figure 8. The block diagram of Madgwick's data fusion algorithm for the 9-axis IMU Sensor.

A 3-axis accelerometer measures not only the magnitude and direction of the gravity in the sensor frame but also linear velocity due to motion of the sensor. A 3-axis magnetometer measures the magnitude and direction of the earth's magnetic field in the sensor frame compounded with distortion. Following the derivation from (22) to (25), the estimated quaternion ${}^{S}_{E}\hat{q}_{acc,mag\_est}(t)$ at time t can be calculate by combining the output data of accelerometer and the magnetometer, we have

$$ {}^{S}_{E}\hat{q}_{acc,mag\_est}(t) = {}^{S}_{E}\hat{q}_{est}(t-1) - \kappa \frac{\nabla f_{acc,mag}}{\|\nabla f_{acc,mag}\|} \tag{26} $$

$$ \nabla f_{acc,mag} = J^{T}_{acc,mag}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{acc,mag}(t)) f({}^{S}\hat{d}_{acc,mag\_est}(t), {}^{S}\hat{S}_{acc,mag}(t)) \tag{27} $$

$$ J^{T}_{acc,mag}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{acc,mag}(t)) = \begin{bmatrix} J^{T}_{acc}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{acc}(t)) \\ J^{T}_{mag}({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{mag}(t)) \end{bmatrix} \tag{28} $$

$$ \begin{aligned} f({}^{S}\hat{d}_{acc,mag\_est}(t), {}^{S}\hat{S}_{acc,mag}(t)) &= \begin{bmatrix} f({}^{S}\hat{d}_{acc\_est}(t), {}^{S}\hat{S}_{acc}(t)) \\ f({}^{S}\hat{d}_{mag\_est}(t), {}^{S}\hat{S}_{mag}(t)) \end{bmatrix} \\ &= \begin{bmatrix} f({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{acc}(t), {}^{S}\hat{S}_{acc}(t)) \\ f({}^{S}_{E}\hat{q}_{est}(t-1), {}^{E}\hat{d}_{mag}(t), {}^{S}\hat{S}_{mag}(t)) \end{bmatrix} \end{aligned} \tag{29} $$

The estimated quaternion ${}^{S}_{E}\hat{q}_{acc,mag\_est}(t)$ is achieved based on a previous estimate of quaternion ${}^{S}_{E}\hat{q}_{est}(t-1)$ and the objective function $\nabla f_{acc,mag}$. This objective function is defined by measured data of accelerometer ${}^{S}\hat{S}_{acc}(t)$ and magnetometer ${}^{S}\hat{S}_{mag}(t)$ in sensor frame.

Using the concept of the complementary filter presented in [31], the final estimated quaternion ${}^S_E\hat{q}_{est}(t)$ is obtained through the fusion of the two estimated quaternions ${}^S_E\hat{q}_{\omega\_est}(t)$ and ${}^S_E\hat{q}_{acc,mag\_est}(t)$,

$$ {}^S_E\hat{q}_{est}(t) = (1-\gamma){}^S_E\hat{q}_{\omega\_est}(t) + (\gamma){}^S_E\hat{q}_{acc,mag\_est}(t) \tag{30}$$

where $(1-\gamma)$ and γ are weights of ${}^S_E\hat{q}_{\omega\_est}(t)$ and ${}^S_E\hat{q}_{acc,mag\_est}(t)$, respectively. The convergence rate of ${}^S_E\hat{q}_{acc,mag\_est}(t)$ is defined as $\frac{\kappa}{\Delta t}$ and the divergence rate of ${}^S_E\hat{q}_{\omega\_est}(t)$ is defined as β . Following the derivation in [26], the optimal value of γ is defined as

$$\gamma = \frac{\beta}{\frac{\kappa}{\Delta t} + \beta} \tag{31}$$

It is assumed that κ is large, (26) and (31) can be rewritten

$$ {}^S_E\hat{q}_{acc,mag\_est}(t) \approx -\kappa\frac{\nabla f_{acc,mag}}{\|\nabla f_{acc,mag}\|} \tag{32}$$

$$\gamma = \frac{\beta\Delta t}{\kappa} \approx 0 \tag{33}$$

Substituting (20), (21), (32) and (33) into (30), we have

$$\begin{aligned} {}^S_E\hat{q}_{est}(t) &= (1-\gamma){}^S_E\hat{q}_{\omega\_est}(t) + (\gamma){}^S_E\hat{q}_{acc,mag\_est}(t) \\ &= (1-0){}^S_E\hat{q}_{\omega\_est}(t) + \frac{\beta\Delta t}{\kappa}(-\kappa\frac{\nabla f_{acc,mag}}{\|\nabla f_{acc,mag}\|}) \\ &= {}^S_E\hat{q}_{est}(t-1) + {}^S_E\dot{\hat{q}}_{\omega\_est}(t)\Delta t - \beta\frac{\nabla f_{acc,mag}}{\|\nabla f_{acc,mag}\|}\Delta t \\ &= {}^S_E\hat{q}_{est}(t-1) + \frac{1}{2}{}^S_E\hat{q}_{est}(t-1)\otimes {}^S\omega(t)\Delta t - \beta\frac{\nabla f_{acc,mag}}{\|\nabla f_{acc,mag}\|}\Delta t \end{aligned} \tag{34}$$

From (34), it can be seen the fusion algorithm minimizes β for the measured magnetic field will be distorted by the presence of ferromagnetic elements in the vicinity of the magnetometer. Due to the error, the estimated direction vector of the earth's magnetic field ${}^E\hat{d}_{mag\_est}(t)$ has nonzero Y-axis component, the ${}^E\hat{d}_{mag\_est}(t)$ is calculated as

$$\begin{aligned} {}^E\hat{d}_{mag\_est}(t) &= [0 \;\; {}^E d_{mag\_x\_est}(t) \;\; {}^E d_{mag\_y\_est}(t) \;\; {}^E d_{mag\_z\_est}(t)] \\ &= {}^S_E\hat{q}_{est}(t-1)\otimes {}^S\hat{d}_{mag}(t)\otimes {}^S_E\hat{q}^*_{est}(t-1) \end{aligned} \tag{35}$$

where ${}^E d_{mag\_y\_est}$ is nonzero Y-axis component in the earth frame. These nonzero Y-axis component can be corrected by normalizing to have only X and Z-axes components of the earth frame [26]. Hence

$$ {}^E\hat{d}_{mag\_comp\_est}(t) = [0 \;\; \sqrt{{}^E d^2_{mag\_x\_est}(t) + {}^E d^2_{mag\_y\_est}(t)} \;\; 0 \;\; {}^E d_{mag\_z\_est}(t)] \tag{36}$$

where ${}^E\hat{d}_{mag\_comp\_est}(t)$ is the compensated magnetic vector in the earth frame. Using this compensated magnetic vector, the gyroscope error can be corrected further.

After estimating the ${}^S_E\hat{q}_{est}(t)$, the Euler's angle can be achieved by using the following equation,

$$ {}^S_E\hat{q}_{est}(t) = [q_1^{'}(t) \;\; q_2^{'}(t) \;\; q_3^{'}(t) \;\; q_4^{'}(t)] \tag{37}$$

$$\psi(t) = \tan^{-1}(\frac{2(q_1^{'}(t)q_4^{'}(t) + q_2^{'}(t)q_3^{'}(t))}{q_1^{'2}(t) + q_2^{'2}(t) - q_3^{'2}(t) - q_4^{'2}(t)}) \tag{38}$$

$$\delta(t) = \sin^{-1}(2q_1^{'}(t)q_3^{'}(t) - 2q_2^{'}(t)q_4^{'}(t)) \tag{39}$$

$$\phi(t) = \tan^{-1}\left(\frac{2(q_1^{'}(t)q_2^{'}(t) + q_3^{'}(t)q_4^{'}(t))}{q_1^{'2}(t) - q_2^{'2}(t) - q_3^{'2}(t) + q_4^{'2}(t)}\right) \tag{40}$$

where $q_1^{'}(t)$, $q_2^{'}(t)$, $q_3^{'}(t)$ and $q_4^{'}(t)$ are components of ${}^S_E\hat{q}_{est}(t)$, $\psi(t)$, $\delta(t)$ and $\phi(t)$ are yaw angle, pitch angle and roll angle of the NWMR, respectively. Because the NWMR moves on the topographic plane, the yaw angle is acted as the direction angle of the robot, therefore the roll angle and pitch angle are omitted.

4.2. Kalman Filter for the GPS Module

The Kalman filter is used to estimate the position of the NWMR and hand-held device from the output data of GPS. This filter is an optimal recursive data processing algorithm to minimize the mean squared error between the actual data and estimated data. The flow chart of Kalman filter is shown in Figure 9, the algorithm works in a two-step process. In prediction step, it predicts the current state variables. In update step, the difference between predicted value and measured value obtained from the sensor readout is taken into consideration. Kalman gain is counted for estimating the new predicted value and new uncertainty variance.

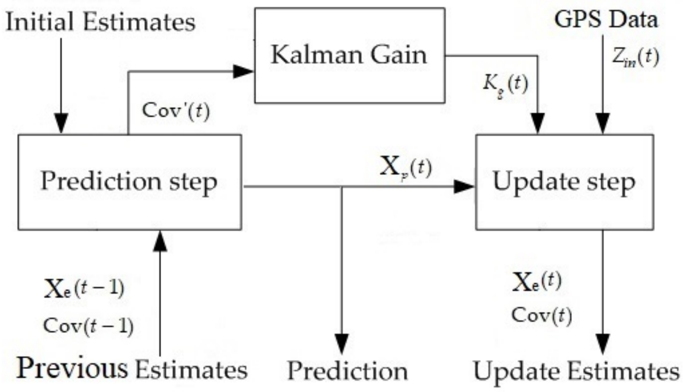


Figure 9. Kalman filter recursive algorithm.

Prediction Step:

$$\begin{aligned} \mathbf{X_p(t)} &= \mathbf{FX_e(t-1)} \\ \mathbf{Cov^{'}(t)} &= \mathbf{FCov(t-1)F^T + Q_{distb}} \end{aligned} \tag{41}$$

where the current state variables matrix $\mathbf{X_p(t)}$ is derived from its previous estimate value $\mathbf{X_e(t-1)}$ and $\mathbf{F}$ is state transient matrix. $\mathbf{Cov^{'}(t)}$ is prior error covariance matrix and $\mathbf{Q_{distb}}$ is the expect value of the system disturbance.

Kalman Gain:

$$\mathbf{K_g(t)} = \frac{\mathbf{Cov^{'}(t)H^T}}{\mathbf{HCov^{'}(t)H^T + R_{error}}} \tag{42}$$

where $\mathbf{K_g(t)}$ is the Kalman gain, $\mathbf{H}$ is the noiseless transition matrix from the real state variables and $\mathbf{R_{error}}$ is the expect value of the error measurement.

Update Step:

$$\begin{aligned} \mathbf{X_e(t)} &= \mathbf{X_p(t) + K_g(t)(Z_{in}(t) - HX_p(t))} \\ \mathbf{Cov(t)} &= \mathbf{(1 - K_g(t)H)Cov^{'}(t)} \end{aligned} \tag{43}$$

where the update estimate $\mathbf{X_e(t)}$ is derived from the measured data $\mathbf{Z_{in}}$. $\mathbf{Cov(t)}$ is the error covariance of the estimate datas.

The advantage of the Kalman filter is that the algorithm uses only the current measured data, the previously calculated state and its error covariance matrix, the past information is not needed. It prevents from the requirement of additional memory for storing the history data.

5. Simulations and Experimental Results

In this section, the simulations and the experiments are illustrated to demonstrate the merit and the effectiveness of the developed NWMR. Section 5.1 describes the hardware of the implemented NWMR and the hand-held device. In Section 5.2, the comparison results for the tracking performance of the controllers proposed by [22,23] are shown via MATLAB software. The experiments of trajectory tracking and dynamic real-time tracking of NWMR are demonstrated in Section 5.3.

5.1. Hardware Implementations

Figure 10a shows the physical control system of the NWMR. Frame marked with number 1 is the core control board of the NWMR; Frame 2 is the power switching board which is used to activate the NWMR and frame 3 is the relay board which is used as brakes to stop the NWMR. Frame 4 is motor driver board and it is used to drive the two wheels of NWMR. Figure 10b shows the lateral view of the core control board.

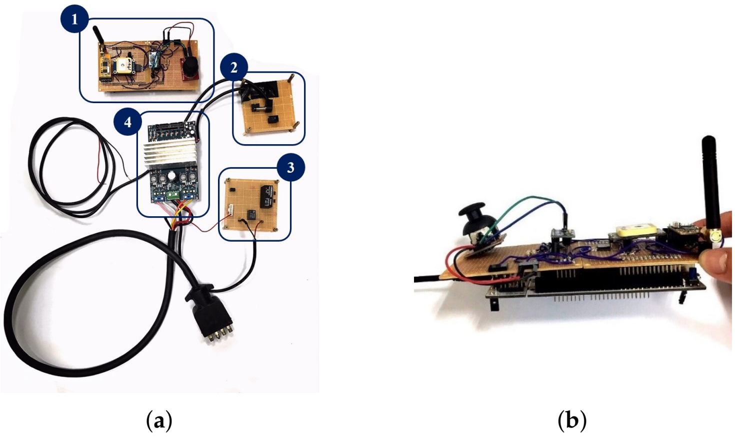


Figure 10. The overall view of the control system of the NWMR: (**a**) The physical control system of the NWMR; (**b**) The lateral view of the core control board of the NWMR.

In the proposed NWMR, the part number MPU-9250 is used for 9-axis IMU, NEO-6M V2 is for GPS module and APC-220 is for RF transceiver. The STM32F429 embedded system is equipped under the core control board. Figure 11a,b shows the mechanism of the NWMR. Frame 1 of Figure 11a is the control system of the NWMR and frame 2 is the motors and battery modules. The hand-held device is shown in Figure 12. The blue bottom is utilized to activate the hand-held device.

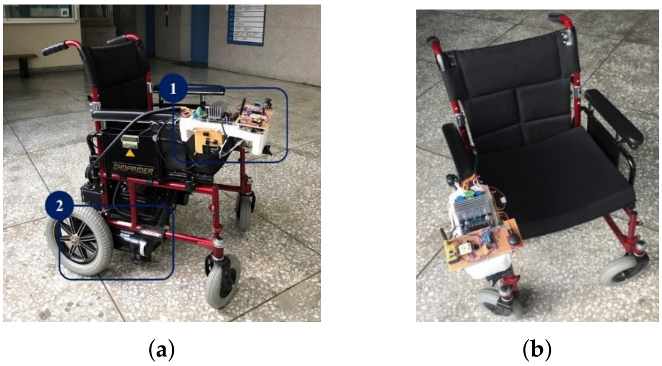


Figure 11. The overall view of the NWMR: (**a**) The lateral view; (**b**) The top view.

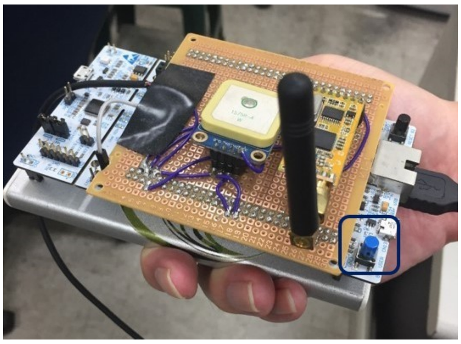


Figure 12. The physical view of the hand-held device.

5.2. Simulation Results

This section describes the simulation results of the tracking controller. For simplicity, the physical phenomenon presented in the real world, i.e., the friction of NWMR, the error range from the GPS modules which are mounted on the hand-held device and the NWMR, respectively, are ignored, therefore, the conceptual comparisons of the tracking capability for the tracking controllers in [22,23] are conducted. Three tracking cases are illustrated to compare the tracking performance of the controllers. First case is straight line tracking with linear velocity and second case is trajectory tracking of the circle with linear velocity and linear angular velocity. Third case is the random dynamic tracking example. As mentioned in Section 3.2, in order to achieve the controller to drive the trajectories asymptotically stable, the parameters of the controller are set as $k_1 = 2$, $k_2 = 4$, $k_3 = 10$ and $k_4 = 2$.

Figure 13 shows the tracking trajectory of the two controllers. The Reference straight line can be treated as the trajectory of the hand-held device. The initial posture of the hand-held device $P_d(0) = [x_d, \ y_d, \ \theta_d]$ is set to $[4, \ 0, \ \frac{\pi}{4}]$, the initial linear velocity $v_d(0)$ is 4 m/s and the angular velocity ω_d is 0 rad/s. The initial posture of the two tracking controllers $P_c(0)$ is set to $[-4, \ 8, \ -\frac{\pi}{4}]$.

Figure 14a shows the posture error x_e in X-axis. We set the convergent point at 0.03275, the figure shows that the convergence time of the controller proposed by [23] is 0.5293 sec, the convergence time of the controller by [22] is 1.891 sec. Figure 14b shows the posture error y_e in Y-axis. By setting the convergent point as -0.00673, the convergence time of 3.177 sec and 3.595 sec can be obtained by using [22,23], respectively.

Figure 14c illustrates the angular error θ_e. For the convergent point 0.004761, the figure shows that the convergence time of the controller in [23] is 1.824 s, the convergence time of the controller in [22] is 3.345 s.

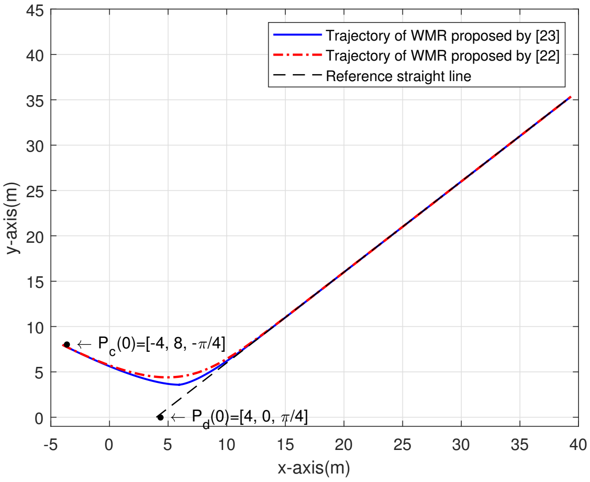


Figure 13. The trajectory of the straight line tracking.

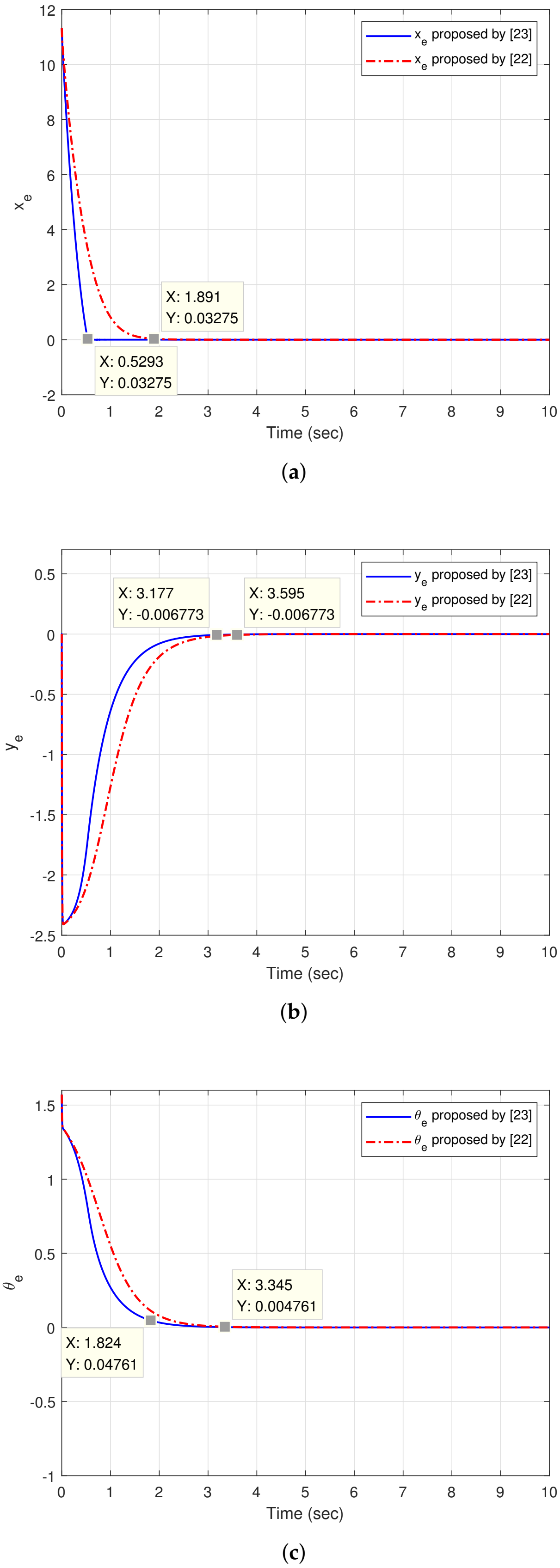


Figure 14. The posture errors of straight line tracking.: (**a**) x_e; (**b**) y_e; (**c**) θ_e.

Figure 15 shows the simulation result of the trajectory tracking of the circle. The initial posture of hand-held device $P_d(0)$ is set as $[4,\ 0,\ \frac{\pi}{5}]$, the initial linear velocity $v_d(0)$ is 5 m/s and the angular velocity ω_d is 1 rad/s. The initial posture of two tracking controllers $P_c(0)$ is set to $[-4,\ -1,\ 0]$. Figure 16a shows the posture error x_e in X-axis. As seen in the figure, for the convergent point -0.01604, the convergence time of the controller [23] is 2.692 s, the convergence time of the controller [22] is 3.01 s. Figure 16b shows the posture error y_e in Y-axis. We set the convergent point at 0.003297, the figure shows that the convergence time of the controller proposed by [23] is 2.952 s, the convergence time of the controller proposed by [22] is 3.811 s. Figure 16c illustrates the angular error θ_e. For the convergent point of -0.0004928, the figure shows that the convergence time of the controller by [23] is 3.969 s, the convergence time of the controller by [22] is 4.328 s.

From the simulation results of straight line tracking and trajectory tracking of the circle, it can readily seen that the controller proposed by [23] can provide a better response than the controller proposed by [22].

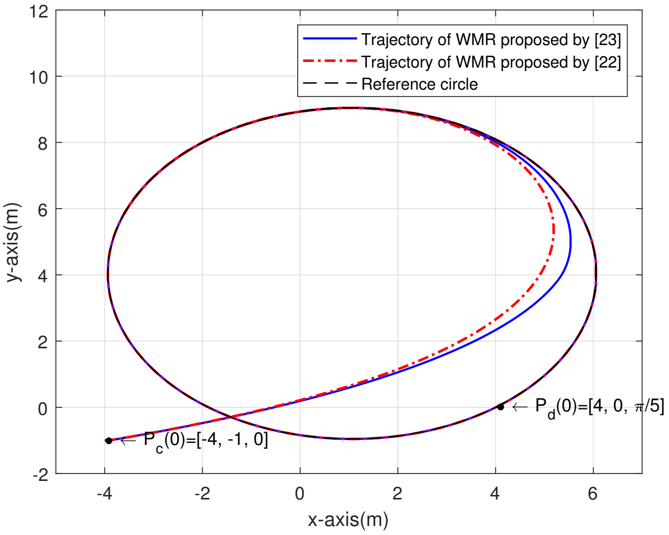


Figure 15. The trajectory tracking of the circle.

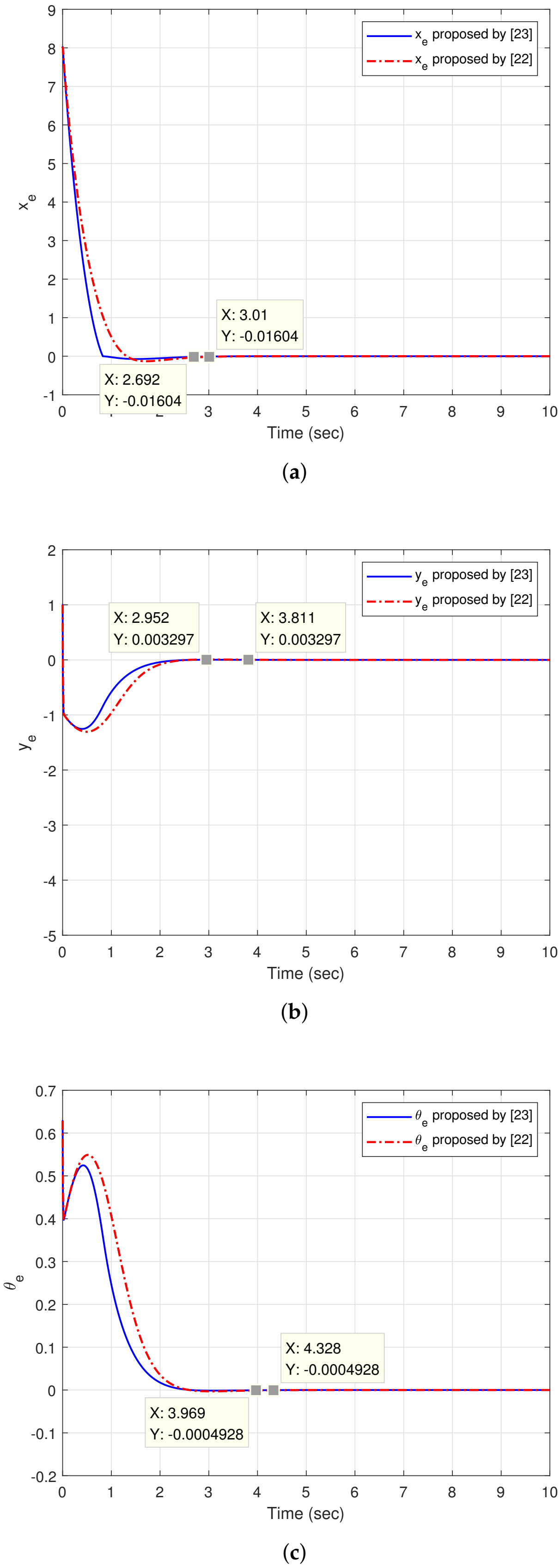
In order to show the dynamic real-time tracking, the θ_d of the tracking object in (5) needs be continuously updated. For the Consideration of the four-quadrant angle conversion, the actual angle θ_{d_update} is recalculated by the following equation,

$$\begin{aligned}
\theta_{d\_update} &= 90^\circ - \theta_d \quad if \;\; 0^\circ < \theta_d < 90^\circ \\
\theta_{d\_update} &= 90^\circ + \theta_d \quad if \;\; 90^\circ < \theta_d < 180^\circ \\
\theta_{d\_update} &= 270^\circ + \theta_d \quad if \;\; 180^\circ < \theta_d < 270^\circ \\
\theta_{d\_update} &= 270^\circ - \theta_d \quad if \;\; 270^\circ < \theta_d < 360^\circ
\end{aligned} \tag{44}$$

The linear velocity of the tracking object u_d in (15) also needs be continuously updated and the actual linear velocity v_{d_update} is

$$v_{d\_update} = \frac{\sqrt{(x_d(t) - x_d(t-1))^2 + (y_d(t) - y_d(t-1))^2}}{\Delta t} \tag{45}$$

where $x_d(t)$ and $x_d(t-1)$ are the positions toward X direction at time t and $t-1$, respectively. $y_d(t)$ and $y_d(t-1)$ are the positions toward Y direction at time t and $t-1$, respectively, Δt is sampled time. Because the tracking object is treated as a point in the world X-Y coordinate, the angular velocity of the hand-held device ω_d in (15) is set to zero for simplicity.



(**a**)

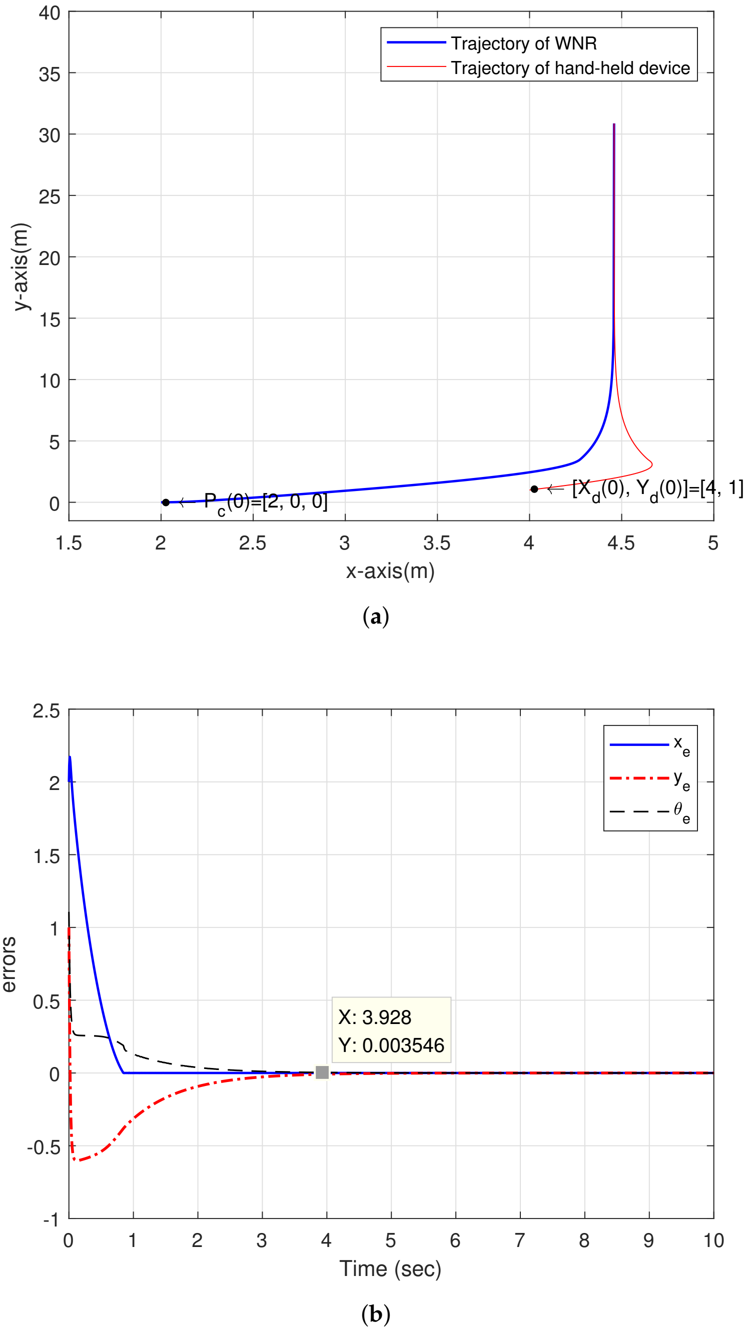
(**b**)

(**c**)

Figure 16. The posture error of trajectory tracking of the circle: (**a**) x_e; (**b**) y_e; (**c**) θ_e.

Figure 17a shows the simulation result of the dynamic real-time tracking of the tracking controller proposed by [23]. The initial position of hand-held device $[x_d,\ y_d]$ is $[4,\ 1]$, the initial linear velocity $v_d(0)$ is 3 m/s and the angular velocity ω_d is 0 rad/s. The initial posture of NWMR $P_c(0)$ is $[2,\ 0,\ 0]$. Figure 17b shows the posture error of the NWMR with respect to the hand-held device, the angular error is converged to 0.003546 after 3.928 s.

In order to implement the NWMR, the tracking controller and algorithms mentioned above are all implemented with C language in MDK-ARM integrated development environment.



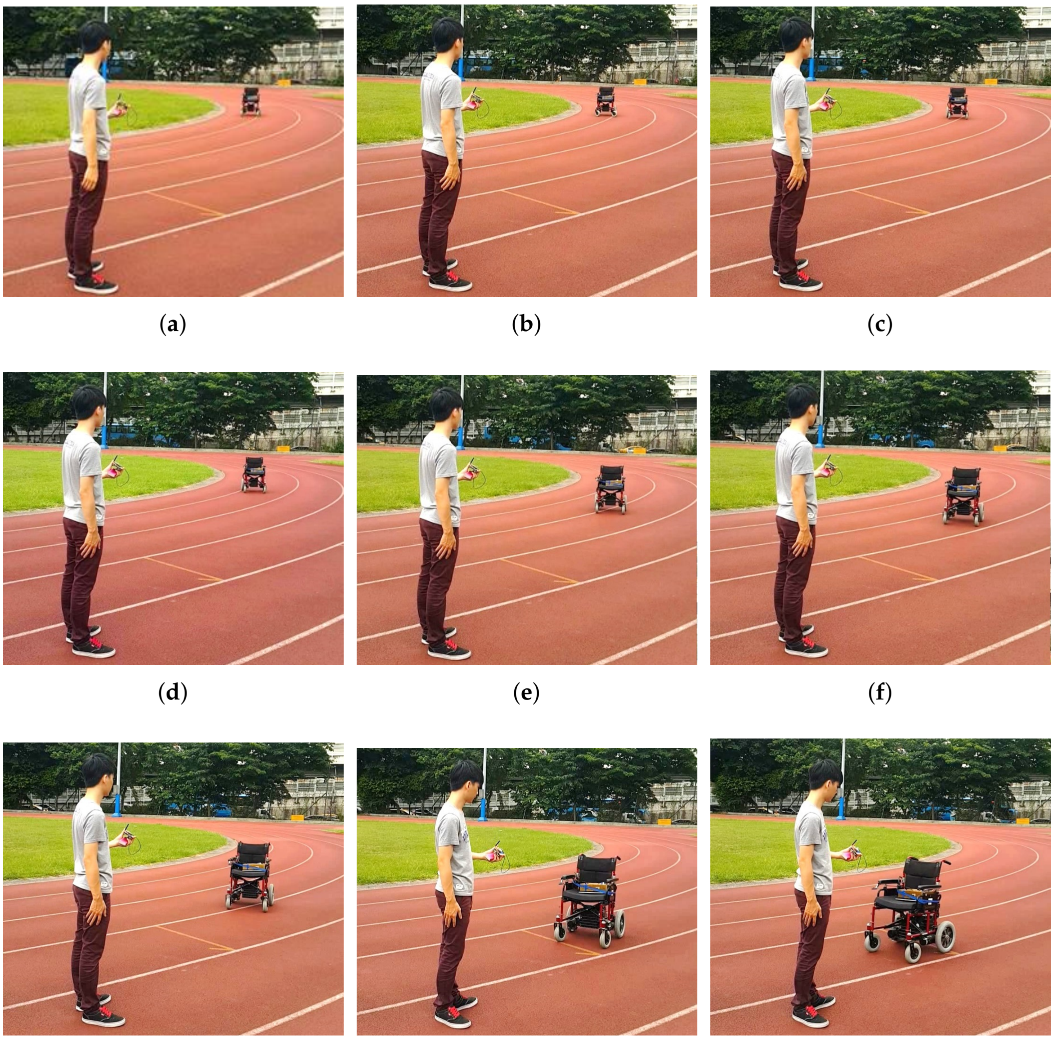
(**a**)

(**b**)

Figure 17. The dynamic real-time tracking: (**a**) Trajectory between NWMR and hand-held device; (**b**) The posture error for the dynamic real-time tracking.

5.3. Experiments

The processes of the trajectory tracking of the NWMR are shown in Figure 18a–i. The video is attached in Supplementary Material 1. We first set the initial postures of NWMR and the hand-held device, the user holds the hand-held device and is acted as tracking object. Figure 18a–i show the trajectory tracking responses of the developed NWMR. The robot moves toward to the user and stop while the distance between the NWMR and objective is smaller than 1.5 m.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 18. The processes of the trajectory tracking.

The dynamic real-time tracking of the NWMR are shown in Figure 19a–l. The video is attached in Supplementary Material 2. These figures show the processes of real-time tracking. From Figure 19a–l, it can be seen that the real-time tracking can be achieved by the developed NWMR. When the user goes forward and go backward, the NWMR can track the hand-held device smoothly. Even if the user turns right and turns left, the robot still can track accurately.

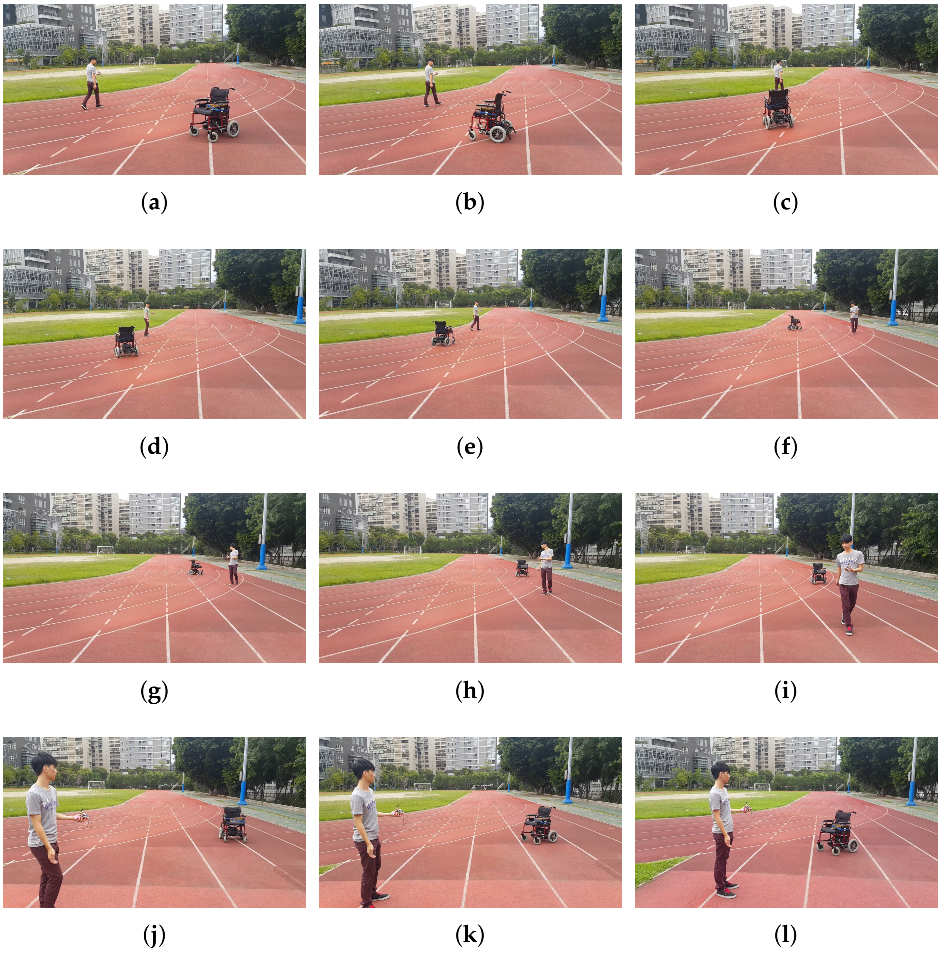


Figure 19. The processes of dynamic real-time tracking.

6. Conclusions and Future Works

In this paper, the trajectory tracking for the NWMR is investigated. GPS and 9-axis IMU sensor are utilized to measure the posture and position of the NWMR and target device. For the consideration of the small posture error and fast convergence in the trajectory tracking, Kalman filter and Madgwick's fusion algorithm are used. The tracking controller with these algorithms are implemented by an embedded system for fast prototype and cost efficiency. The simulations and the experiment results show the feasibility and the effectiveness of the proposed NWMR.

For designing the next generation of the NWMR, obstacle avoidance, pattern recognition, multi-sensor fusion algorithms and power management strategy are four main research topics. For the outdoor application, a common GPS will be replaced by the RTK-GPS due to error range from 20 cm to 2 m. For the obstacle avoidance approach, millimeter-Wave (mmWave) radar sensor and Light-detection and ranging (LIDAR) sensor will be used. The advantage of the mmWave radar sensor is impervious to environmental conditions such as rain, fog and dust. The LIDAR sensor has the advantage of high resolution for a short distance measurement. Pattern recognition module integrated image sensor, optical lens and image processor is used to identify objects by utilizing the machine learning algorithms. Moreover, in order to extend the operating time of NWMR, power management strategy and will be studied further.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/20/24/7055/s1 .

Author Contributions: Conceptualization, S.-H.T.; Circuit implementation, H.-Y.L. and T.-C.L.; Embedded program implementation, L.-H.K. ; Matlab program, L.-H.K.; Validation, Y.-L.S.; Writing draft, S.-H.T. and H.-Y.L.; Math Review, S.-H.T. and Y.-L.S.; Funding, L.-M.C.; Literature Review, H.-Y.L. and L.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology of Taiwan, R.O.C., under Grant MOST-109-2221-E-002-023-, MOST-108-3116-F-027-003-CC2, MOST-109-2221-E-027-078- and MOST-109-2811-E-002-556.

Acknowledgments: The authors also gratefully acknowledge the helpful comments and suggestions of the Editor and anonymous reviewers, which have improved the presentation of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ullrich, G . *Automated Guided Vehicle Systems*; Springer: Berlin/Heidelberg, Germany, 2015.
2. Luo, J.; Ni, H.; Zhou, M. Control Program Design for Automated Guided Vehicle Systems via Petri Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 44–55.
3. Kang, M.-C.; Kim, K.-S.; Noh, D.-K.; Han, J.-W.; Ko, S.-J. A Robust Obstacle Detection Method for Robotic Vacuum Cleaners. *IEEE Trans. Consum. Electron.* **2014**, *60*, 587–595. [CrossRef]
4. Chou, C.-Y; Juang, C.-F. Navigation of an Autonomous Wheeled Robot in Unknown Environments Based on Evolutionary Fuzzy Control. *Inventions* **2018**, *3*, 3. [CrossRef]
5. Leal, J.C.E.; Torres, J.G.R.; Tello, E.R.; Angulo, J.R.M. Multi-robot Exploration Using Self-Biddings under Constraints on Communication Range *IEEE Latin Am. Trans.* **2014**, *60*, 587–595.
6. Borraz, R.; Navarro, P.J.; Fernández, C.; Alcover, P.M. Cloud Incubator Car: A Reliable Platform for Autonomous Driving. *Appl. Sci.* **2018**, *8*, 303. [CrossRef]
7. Elangovan, K.; Tamilselvam, Y.K.; Mohan, R.E.; Iwase, M.; Takuma, N.; Wood, K.L.; Alcover, P.M. Fault Diagnosis of a Reconfigurable Crawling-Rolling Robot Based on Support Machines. *Appl. Sci.* **2017**, *7*, 1025. [CrossRef]
8. Höflinger, F.; Müller, J.; Zhang, R.; Reindl, L.M.; Burgard, W. A Wireless Micro Inertial Measurement Unit (IMU). *IEEE Trans. Instrum. Meas.* **2013**, *62*, 2583–2595. [CrossRef]
9. Velázquez, R.; Pissaloux, E.; Rodrigo, P.; Carrasco, M.; Giannoccaro, N.I. Lay-Ekuakille, An Outdoor Navigation System for Blind Pedestrians Using GPS and Tactile-Foot Feedback. *Appl. Sci.* **2018**, *8*, 578. [CrossRef]
10. Knoop, V.L.; de Bakker, P.F.; Tiberius, C.C.J.M.; Arem, B.V. Lane Determination With GPS Precise Point Positioning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2503–2513. [CrossRef]
11. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Appl. Sci.* **2018**, *8*, 231. [CrossRef]
12. Rotondo, D.; Puig, V.; Nejjari, F. A Fault-Hiding Approach for the Switching Quasi-LPV Fault-Tolerant Control of a Four-Wheeled Omnidirectional Mobile Robot. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3932–3944. [CrossRef]
13. Kim, H.; Kim, B. K. Online Minimum-Energy Trajectory Planning and Control on a Straight-Line Path for Three-Wheeled Omnidirectional Mobile Robots. *IEEE Trans. Ind. Electron.* **2014**, *61*, 4771–4779. [CrossRef]
14. Batayneh, W.; AbuRmaileh, Y. Decentralized Motion Control for Omnidirectional Wheelchair Tracking Error Elimination Using PD-Fuzzy-P and G-PID controllers. *Sensors* **2020**, *20*, 3525. [CrossRef] [PubMed]
15. Kim, C.; Suh, J.; Han, J.-H. Development of a Hybrid Path Planning Algorithm and a Bio-Inspired Control for an Omni-Wheel Mobile Robot. *Sensors* **2020**, *20*, 4258. [CrossRef] [PubMed]
16. Reina, G. Cross-Coupled Control for All-Terrain Rovers. *Sensors* **2013**, *13*, 785–800. [CrossRef] [PubMed]
17. Sun, W.; Tang, S.; Gao, H. Two Time-Scale Tracking control of Nonholonomic Wheeled Mobile Robots. *IEEE Control Syst. Technol.* **2016**, *24*, 2059–2069. [CrossRef]
18. Yang, H.; Fan, X.; Shi, P.; Hua, C. Nonlinear Control for Tracking and Obstacle Avoidance of a Wheeled Mobile Robot With Nonholonomic Constraint. *IEEE Control Syst. Technol.* **2016**, *24*, 741–746. [CrossRef]
19. Rayguru, M.M.; Elara, M.R.; Ramalingam, B.; Muthugala, M.A.; Samarakoon, S.M.B.P. A Path Tracking Strategy for Car Like Robots with Sensor Unpredictability and Measurement Errors. *Sensors* **2020**, *20*, 3077. [CrossRef]
20. He, Z.; Nie, L.; Yin, Z.; Huang, S. A Two-Layer Controller for Lateral Path Tracking Control of Autnomous Vehicles. *Sensors* **2020**, *20*, 3689. [CrossRef]

21. Wang, Z.; Li, G.; Chen, X.; Zhang, H.; Chen, Q. Simultaneous Stabilization and Tracking of Nonholonomic WMRs with Input Constraints: Controller Design and Experimental Validation. *IEEE Trans. Industrial Electron.* **2019**, *66*, 5343–5352. [CrossRef]
22. Kanayama, Y.; Kimura, Y.; Miyazaki, F.; Noguchi, T. A Stable Tracking Control Method for an Autonomous Mobile Robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, Oh, USA, 13–18 May 1990; pp. 384–389.
23. Sanhoury, I.M.H.; Amin, S.H.M.; Husain, A.R. Tracking Control of a Nonholonomic Wheeled Mobile Robot. *Precis. Instrum. Mechanol.* **2012**, *1*, 7–11.
24. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *ASME J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
25. Basith, A.L.; Arifin, A.; Arrofiqi, F.; Watanabe, T.; Nuh, M. Embedded fuzzy logic controller for functional electrical stimulation system. In Proceedings of the International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, Indonesia, 28–30 July 2016; pp. 89–94.
26. Madgwick, S.O.H.; Harrison, A.J.L.; Vaidyanathan, R. Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm. In Proceedings of the IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, 29 June–1 July 2011; pp. 1–7.
27. Admiraal, M.; Wilson, S.; Vaidyanathan, R. Improved Formulation of the IMU and MARG Orientation Gradient Descent Algorithm for Motion Tracking in Human-Machine Interfaces. In Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent System (MFI 2017), Daegu, Korea, 16–18 November 2017; pp. 403–410.
28. Yang, J.-M.; Kim, J.-H. Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots. *IEEE Trans. Robot. Autom.* **1999**, *15*, 578–587. [CrossRef]
29. Pathirana, P.N.; Karunarathne, M.S.; Williams, G.L.; Nam, P.T.; Durrant-Whyte, H. Robust and Accurate Capture of Human Joint Pose Using an Inertial Sensor. *IEEE J. Transl. Eng. Health Med.* **2018**, *6*, 1–11. [CrossRef] [PubMed]
30. Park, F.C.; Ravani, B. Smooth invariant interpolation of rotations. *ACM Trans. Graph.* **1997**, *16*, 277–295. [CrossRef]
31. Mahony, R.; Hamel, T.; Pflimlin, J.-M. Nolinear Complementary Filters on the Special Orthogonal Group. *IEEE Trans. Autom. Control* **2008**, *53*, 1203–1218. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.