Differential Flatness-based Kinematic and Dynamic Control of a Differentially Driven Wheeled Mobile Robot

Chin Pei Tang, *IEEE Member*

***Abstract*— In this paper, we present an integrated motion planning and control framework for control of a differentially driven wheeled mobile robot based on the differential flatness property exhibits in the models of the system. The goals of the paper are: (a) to present a simple online control method for wheeled mobile robot using both its kinematic and dynamic models (with the emphasis on dynamic model that has not been studied extensively), and (b) to illustrate the straight-forward manner of the controller design for easy implementation. It can be shown that the extension of the control from the kinematic to the dynamic model is straight-forward using the same flat outputs under the same diffeomorphism between the state/input space and the flat output space. Given that the application of such system is considerably ubiquitous recently, this paper is a reminder that a simple control method exists without too much effort in developing new dedicated controllers. The paper also serves as guide for practitioners who look for a straightforward lower-level robot controller so that they can concentrate on the higher level motion planning design.**

I. Introduction

Wheeled Mobile Robots (WMR) have been proven extremely useful in a large variety of robotics applications, especially when mobility is of the high priority during the task execution. The critical applications can range from industrial settings, to military systems, to home robotics, to even education robotics. In fact, such system has been becoming more ubiquitous recently since there have been many commercially available WMR in the market today, for instance, iRobot [1], Khepera [2], AmigoBot [3], Garcia [4], Lego Mindstorms [5], Segway [6], Kiva Systems [7], to name a few. Some of them have been extensively implemented in the academic research groups, especially in the multi-robot context. More interestingly, some of them have even been successfully deployed in real engineering applications.

The control of WMR is used to be very difficult due to the non-integrable velocity constraints termed "NonHolonomic (NH) constraints" imposed by the wheels - see Chap. 12 of [8]. While such nontrivial constraint has been extensively studied in the past few decades, most of the controller designs still require considerable understanding of the mathematics behind the mechanisms. Based on the author's experience, to computer scientists or electrical engineers, who focus more on the higher level planning or artificial intelligent designs, having them to understand such mathematical concepts behind the mechanics can be a challenge. They need to spend considerable effort just to get the robots to track a desired trajectory, and often attempt to stay away from these mechanics. However, there is an emerging need of tight integration of various disciplines to design the next generation Cyber-Physical Systems (CPS) - this paper forms an effort in bridging this gap. One of the major goal of this paper is to illustrate how *simple* and straightforward a controller design for such system in the lower level can be by exploiting the Differential Flatness (DF) property in both the (first-order) kinematic and (second-order) dynamic models of a WMR. It is believed that such controller can attack a large class of WMR systems with considerably wide range of applications, as the required task is mainly to track the Cartesian coordinates of the center of the wheel axle of the WMR.

To succinctly summarize, by virtue of the DF approach, the states and inputs can be parameterized by a finite set of independent variables, called the flat outputs, and their time derivatives. Moreover, the number of flat outputs is equal to the number of control inputs. This enables the transformation of nonlinear differential equations into a system of algebraic equations which are generally simpler to solve. Hence, DF can be very useful for trajectory planning problem since the desired trajectory can be planned in flat output space algebraically using a variety of interpolating functions, including polynomials of appropriate order, to match terminal conditions. In addition, exponential stabilizing controllers can be developed since the system has the representation of a chain of integrators in the flat output space. These features are very attractive since the generation of *feasible* trajectories for general NH systems is difficult in the state space.

There has also been a lot of effort concentrating on the design of new structures for WMR to improve the instantaneous mobility, for instance using Swedish wheels or different mechanisms to achieve instantaneous omnidirectional motion - see [9]. However, the emphasis of this paper is to attack the most popular configuration of Differentially-Driven WMR (DD-WMR), i.e. two wheels mounted colinearly on the left and right side of the chassis of the robot to provide motion. In fact, most of the robots listed in the above examples are equipped with such structural configuration, so the method presented is considerably useful. Furthermore, such DF property can also be exploited in various WMR systems with different configurations [10].

It is also important to note that the work of exploiting the DF properties in the full dynamic WMR model is quite limited. The consideration of full dynamic model is very important when the inertial properties of the robot is not

This work was supported in part by NSF CAREER Award Grant IIS-0347653 and UT Dallas Research Fund.

C. P. Tang is currently with Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA email: chinpei@ieee.org

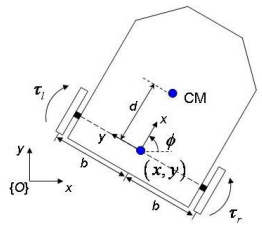

Fig. 1. Schematic of differentially-driven wheeled mobile robot (DD-WMR)

negligible and/or significantly affect the performance of the controller, or when there are needs in high speed or difficult terrain operations that require force models. There are some dedicated papers that propose control methods for a full dynamic WMR model. Sarkar *et. al.* [11] show that the WMR is not input-state linearizable, but is input-output linearizable if the *output* is not chosen to be the center of the wheel-axle. Fierro and Lewis [12] present a control method to backstep from a kinematic model to the full dynamic model. However, in this paper, we show that if the flat outputs are chosen to be the Cartesian coordinates of the center of the wheel axle, an integrated motion planning and control method can be extended from the kinematic model to the full dynamic model in the straight-forward manner with minimal modification. We also present sensitivity studies that show the effectiveness of the control to account for the realistic uncertainties in the dynamic parameters. Hence, the contribution of this paper comes from *the presentation of a simple integrated motion planning and control method for the control of a DD-WMR for both its kinematic and dynamic models by exploiting its differential flatness property.* It is believed that it will simplify considerably the tedious effort for the practitioner, who concentrates more on the high level control design.

II. System Description

Referring to Fig. 1, the DD-WMR is actuated by the right and left wheels with radii r located at a distance b from the center of the wheel axle with Cartesian coordinates (x,y), and the location of the center of mass (CM) is d away from this center. The inputs to the system are wheel velocities $\dot{\theta}_r$ and $\dot{\theta}_l$ in the kinematic model, or τ_r and τ_l in the dynamic model. The subscripts r and l refers respectively to the quantities of right and left wheels. The configuration of the DD-WMR can be completely described by the generalized coordinates of $q = [x,y,\phi]^T$, where ϕ is the heading orientation of the robot.

A. Kinematic Model

The kinematic (unicycle) model of the WMR can be written as follows differential equations [8]:

$$\dot{x} = v\cos\phi, \dot{y} = v\cos\phi, \dot{\phi} = \omega \tag{1}$$

where v and ω are respectively the forward and angular velocities of the WMR. Eq. (1) can be written in the matrix form as:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S\nu \tag{2}$$

Hence, given the input ν, we can compute the corresponding generalized velocities $\dot{q}$ of the WMR. The operator S is the null-space of the NH constraint $A = [\sin\phi, -\cos\phi, 0]$ that produces the *feasible motion* through the inputs. The corresponding wheel inputs can then be computed by:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

B. Dynamic Model

The *constrained* dynamic model can be derived using Euler-Lagrange method [8] as:

$$M\ddot{q} + C\dot{q} = E\tau - A^T\lambda \tag{4}$$

where:

$$M = \begin{bmatrix} m_0 & 0 & -dm_0\sin\phi \\ 0 & m_0 & dm_0\cos\phi \\ -dm_0\sin\phi & dm_0\cos\phi & d^2m_0 + I_0 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 0 & -dm_0\dot{\phi}\cos\phi \\ 0 & 0 & -dm_0\dot{\phi}\sin\phi \\ 0 & 0 & 0 \end{bmatrix}, \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix},$$

and M is the mass matrix, C is the Coriolis/centrifugal matrix, τ is the torque input vector, λ is the constrained force due to the NH constraint, and m_0 and I_0 are respectively the mass and moment of inertia of the WMR about its CM. Note that we do not include the mass and the moment of inertia of the wheels, which is different from [11], [12]. This is practical since they are often difficult to measure but negligible. The actuation matrix E is then needed to compute differently. Invert Eq. (3) to form:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r}{2}\begin{bmatrix} 1 & 1 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \tag{5}$$

Substitute Eq. (5) into Eq. (2), and define:

$$K = \frac{r}{2}\begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} = \frac{r}{2}\begin{bmatrix} \cos\phi & \cos\phi \\ \sin\phi & \sin\phi \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

Due to the Principle of Virtual Work:

$$E = K(K^TK)^{-1} = \frac{1}{r}\begin{bmatrix} \cos\phi & \cos\phi \\ \sin\phi & \sin\phi \\ b & -b \end{bmatrix} \tag{6}$$

Finally, to convert Eq. (4) to the unconstrained form, i.e. eliminate λ term, we take $\nu = [v,\omega]^T$ as the minimal projected coordinates, and pre-multiply both sides by S^T defined in Eq. (2). Note that $S^TA^T = 0$, the projected dynamic model can then be written in the state space (SS) form as:

$$\dot{q} = S\nu, \dot{\nu} = f(q) + g(q)\tau \tag{7}$$

where:

$$\bar{M} = S^T M S = \begin{bmatrix} m_0 & 0 \\ 0 & d^2 m_0 + I_0 \end{bmatrix}$$
$$\bar{V} = S^T \left(M \dot{S} \nu + C \dot{q} \right) = \begin{bmatrix} -d m_0 \dot{\phi}^2 \\ d m_0 \nu \dot{\phi} \end{bmatrix}$$
$$\bar{E} = S^T E = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ b & -b \end{bmatrix}$$

and $f(q) = -\bar{M}^{-1}\bar{V}$, $g(q) = \bar{M}^{-1}\bar{E}$.

III. Differential Flatness

A. Kinematic Model

It can be shown that Eq. (1) is not statically feedback linearizable. Define the flat outputs:

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{8}$$

and differentiate with respect to (wrt) time, and from Eq. (1):

$$\dot{F} = \begin{bmatrix} \dot{F}_1 \\ \dot{F}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} \tag{9}$$

The mapping between the inputs to the flat outputs is singular. Introduce *input prolongation* of ν by extending it to a new input. The *prolonged* SS system is now:

$$\dot{x} = \nu\cos\phi, \dot{y} = \nu\sin\phi, \boxed{\dot{\nu} = \eta_\nu}, \dot{\phi} = \omega \tag{10}$$

where η_ν is the new input to the system. To form the feedback linearizable form, we again differentiate Eq. (9) wrt time:

$$\ddot{F} = \begin{bmatrix} \ddot{F}_1 \\ \ddot{F}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\nu\sin\phi \\ \sin\phi & \nu\cos\phi \end{bmatrix} \begin{bmatrix} \eta_\nu \\ \omega \end{bmatrix} \tag{11}$$

$\ddot{F}$ is now linear wrt the modified inputs η_ν and ω except when $\nu = 0$.

B. Dynamic Model

Eq. (7) is again not statically feedback linearizable. First introduce an input transformation by taking:

$$\gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = f(q) + g(q)\tau \tag{12}$$

Eq. (7) can then be written as:

$$\dot{q} = S\nu, \dot{\nu} = \gamma \tag{13}$$

Note that $\gamma_1 = \eta_\nu$ from Eq. (10). We introduce the prolongation of η_ν to get the new *prolonged* SS system of:

$$\dot{q} = S\nu, \dot{\nu} = \eta_\nu, \boxed{\dot{\eta}_\nu = \hat{\eta}_\nu}, \ddot{\phi} = \gamma_2 \tag{14}$$

where, again, $\hat{\eta}_\nu$ is the new input to the system. To form the feedback linearizable form, we differentiate Eq. (11) wrt time:

$$\dddot{F} = \begin{bmatrix} \dddot{F}_1 \\ \dddot{F}_2 \end{bmatrix} = \begin{bmatrix} -2\eta_\nu\dot{\phi}\sin\phi - \nu\dot{\phi}^2\cos\phi \\ 2\eta_\nu\dot{\phi}\cos\phi - \nu\dot{\phi}^2\sin\phi \end{bmatrix} + \begin{bmatrix} \cos\phi & -\nu\sin\phi \\ \sin\phi & \nu\cos\phi \end{bmatrix} \begin{bmatrix} \hat{\eta}_\nu \\ \gamma_2 \end{bmatrix} \tag{15}$$

$\dddot{F}$ is now linear wrt the modified inputs $\hat{\eta}_\nu$ and γ_2, except when $\nu = 0$. These process effectively allows us to derive a linear control law in the flat output space in both kinematic and dynamic models with the same flat outputs.

C. Diffeomorphism

We can now establish the one-to-one relationship between the state space and the flat output space. By choosing the flat outputs in Eq. (8), all the state/input variables can be expressed in terms of the flat outputs and their derivatives by:

$$\boxed{\begin{aligned} &x = F_1, y = F_2, \phi = \tan^{-1}\left(\frac{\dot{F}_1}{\dot{F}_2}\right), \nu = \sqrt{\dot{F}_1^2 + \dot{F}_2^2} \\ &\dot{\nu} = \frac{\dot{F}_1\ddot{F}_1 + \dot{F}_2\ddot{F}_2}{\sqrt{\dot{F}_1^2 + \dot{F}_2^2}}, \dot{\phi} = \frac{\dot{F}_1\ddot{F}_2 - \ddot{F}_1\dot{F}_2}{\dot{F}_1^2 + \dot{F}_2^2} \end{aligned}} \tag{16}$$

Conversely, the states and the inputs can also be expressed completely by the flat outputs (and their time derivatives) as:

$$\boxed{\begin{aligned} &F_1 = x, F_2 = y \\ &\dot{F}_1 = \dot{x} = \nu\cos\phi, \dot{F}_2 = \dot{y} = \nu\sin\phi \\ &\ddot{F}_1 = \dot{\nu}\cos\phi - \nu\dot{\phi}\sin\phi = \eta_\nu\cos\phi - \nu\dot{\phi}\sin\phi \\ &\ddot{F}_2 = \dot{\nu}\sin\phi - \nu\dot{\phi}\cos\phi = \eta_\nu\sin\phi - \nu\dot{\phi}\cos\phi \end{aligned}} \tag{17}$$

IV. Point-to-Point Motion Planning and Control

Given the diffeomorphism in Eqs. (16) and (17), we can now propose a point-to-point motion planning method that takes advantage of this property. Given the terminal conditions:

$$x(t_0), y(t_0), \phi(t_0), \nu(t_0), \dot{\nu}(t_0), \dot{\phi}(t_0),$$
$$x(t_f), y(t_f), \phi(t_f), \nu(t_f), \dot{\nu}(t_f), \dot{\phi}(t_f)$$

where t_0 and t_f are respectively initial and final times, they can be transformed to the corresponding terminal conditions in the flat outputs of:

$$F_1(t_0), \dot{F}_1(t_0), \ddot{F}_1(t_0), F_2(t_0), \dot{F}_2(t_0), \ddot{F}_2(t_0),$$
$$F_1(t_f), \dot{F}_1(t_f), \ddot{F}_1(t_f), F_2(t_f), \dot{F}_2(t_f), \ddot{F}_2(t_f)$$

Hence, any arbitrary trajectories of $F_1(t)$ and $F_2(t)$ can be constructed in the flat output space, provided that each of the trajectory satisfies the terminal conditions. In this paper, we present a simple polynomial-based trajectory planning for a set of given terminal conditions. We choose the polynomials with appropriate orders such that the coefficients can be uniquely determined by the terminal conditions. Specifically, since we have 6 terminal conditions for each flat output, we choose the following fifth order polynomials:

$$\begin{aligned} F_1(t) &= a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \\ F_2(t) &= b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0 \end{aligned} \tag{18}$$

where $a_j, b_j,\ j = 0, \cdots, 5$ are the coefficients that can be uniquely determined using the 6 terminal conditions for each flat outputs. We are now ready to present the control method for both kinematic and dynamic models to achieve these desired trajectories.

A. Kinematic Model

Given Eq. (11), by introducing new inputs $\xi = [\xi_1, \xi_2]^T$, the system can be linearized into the following chain form:

$$\ddot{F} = \xi \tag{19}$$

Let $F_1^d(t)$ and $F_2^d(t)$ are respectively the desired trajectories for the flat output of $F_1(t)$ and $F_2(t)$, and define the errors $\varepsilon_1 = F_1^d - F_1$, $\varepsilon_2 = F_2^d - F_2$. Choosing the following feedback control laws for each input:

$$\begin{aligned} \xi_1 &= \ddot{F}_1^d + p_{k1}\dot{\varepsilon}_1 + p_{k0}\varepsilon_1 \\ \xi_2 &= \ddot{F}_2^d + q_{k1}\dot{\varepsilon}_2 + q_{k0}\varepsilon_2 \end{aligned} \tag{20}$$

where p_{km}, q_{km}, $m = 0, 1$ are the control gains. By applying the control laws in Eq. (20) into Eq. (19), we can obtain the error dynamics of the closed-loop system as:

$$\begin{aligned} \ddot{\varepsilon}_1 + p_{k1}\dot{\varepsilon}_1 + p_{k0}\varepsilon_1 = 0 \\ \ddot{\varepsilon}_2 + q_{k1}\dot{\varepsilon}_2 + q_{k0}\varepsilon_2 = 0 \end{aligned} \tag{21}$$

Appropriately chosen gains can then ensure the error dynamics to be exponentially stable. To compute the corresponding wheel velocity inputs to the system, we first substitute Eq. (20) into Eq. (19), one can then compute the original inputs η_v and ω by inverting Eq. (11). η_v can then be integrated to obtain v. Finally, given v and ω, the input wheel velocity required to control the original system can be computed from Eq. (3).

B. Dynamic Model

Similarly, given Eq. (15), by introducing new inputs $\zeta = [\zeta_1, \zeta_2]^T$, the system can be linearized into the following chain form:

$$\dddot{F} = \zeta \tag{22}$$

Again, choosing the following feedback control laws for each input:

$$\begin{aligned} \zeta_1 &= \dddot{F}_1^d + p_{d2}\ddot{\varepsilon}_1 + p_{d1}\dot{\varepsilon}_1 + p_{d0}\varepsilon_1 \\ \zeta_2 &= \dddot{F}_2^d + q_{d2}\ddot{\varepsilon}_2 + q_{d1}\dot{\varepsilon}_2 + q_{d0}\varepsilon_2 \end{aligned} \tag{23}$$

and p_{dn}, q_{dn}, $n = 0, 1, 2$ are the control gains. We can again show that the error dynamic exponentially stable by appropriately chosen gains. To compute the corresponding wheel torque inputs, we first substitute Eq. (23) into Eq. (22), one can then compute the original inputs $\hat{\eta}_v$ and γ_2 by inverting Eq. (15). $\hat{\eta}_v$ can then be integrated to obtain η_v. Finally, given γ, the input torque required to control the original system can be computed from Eq. (12).

V. Real-Time Simulation Studies

We perform real-time simulations to show the effectiveness of the controllers under the MATLAB Real-Time Windows Target [13] using a Dell Latitude D620 laptop. The sampling time is $5ms$ using `ode1` Euler solver for the model update. These settings are considerably close to low level real-time hardware implementation.

A. Case I: Terminal Conditions along the NH Direction

One of the classical problems in NH system is to require a WMR to parallel park itself. Hence, in this first simulation study, we show the simplicity of using the DF formulation as the integrated motion planning and control tool for this case. We specify the terminal conditions as shown in Table I. The positions and velocities results are plotted in Fig. 2(a) and (b), respectively. We can see that the trajectory is not exactly the usual "parallel parking maneuver" due to the polynomial fitting. However, the solution is such that the robot is always moving in the forward direction, which may be favorable in some implementation. More importantly, the interpolated polynomial trajectories satisfy the NH constraint. Hence, this shows how simple it is in planning and controlling such system even in a dynamic setting. Finally, note that all the simulations are done using the dynamic models since the kinematic case is considerably trivial. We take $m_0 = 20kg$ and $I_0 = 0.3kgm^2$. Fig. 2(c) shows the input torque profiles of the system to create the maneuver, which is important when taking into account for hardware actuation capability.

B. Case II: Error Compensation

In the second case studies, we plan the trajectories based on the terminal conditions listed in Table I. The WMR is required to compensate for the initial error. The control gains are chosen to be reasonably slow, where $p_{d2} = 3$, $p_{d1} = 3$, $p_{d0} = 1$ and $q_{d2} = d3$, $q_{d1} = 3$, $q_{d0} = 1$. We perform a series of studies where there exist initial error in all x , y and ϕ, and they can successfully converge to the planned trajectory. Fig. 3 (a), (b) and (c) show the position, velocity and the corresponding input torque profile when there is an initial error of $\Delta x = 10cm$ and $\Delta y = 10cm$. Again, only dynamic case is considered, and it is important to note that based on the chosen gains the required input torque profiles do not change too much. This is important since we have to operate the system such that it exceed the input saturation.

VI. Sensitivity to Model Parameters

Although the tracking performance of the controllers can be evaluated by sight, some form of quantitative measures would be helpful in evaluating the performance effectively. Three forms of performance measures have been proposed: (a) root square mean error (RSME), (b) maximum of absolute error (MAE), and (c) end point absolute error (EPAE). The most straight-forward measure of the controller performance is to evaluate its average deviation from the desired trajectory. The measure can be described as:

$$RSME = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\| f_i - f_i^d \right\|^2} \tag{24}$$

where N is the number of samples between the time interval, and f_i and f_i^d are the i^{th} values of the actual and desired quantities of interest respectively. The RSME measure is averaging the entire tracking history, which might not be able to capture the intermediate performance within the time interval. Hence, the idea of "overshoot" can be implemented,

where the maximum absolute error values within the time interval can be described as:

$$MAE = \max_i \left\| f_i - f_i^d \right\| \tag{25}$$

where $i = 1, \cdots, N$. The problem of using the MAE measure is that the maximum value might just be the initial error introduced to the system. A "settling-time" motivated measure can also be used, but it is quite involved in the implementation. However, another simpler and more straightforward measure that can be used is to evaluate whether or not the controller is able to track the actual trajectory to the desired end point ($i = N$):

$$EPAE = \left\| f_N - f_N^d \right\| \tag{26}$$

A. Case I: Inertial Property Uncertainties

We take the Simulation Case II as the benchmark in the following studies, and introduce the model uncertainties to m_0 and I_0, varying from 20% to 200%. We feel that these uncertainties would highly affect the performance of the controller. While we can examine tracking performances based on all the performance measures for all the 5 states x, y, ϕ, v, $\dot{\phi}$, however, in this paper we look at only the x and y quantities since they are the most important quantities. The RSME and EPAE of x and y of the controller under such uncertainties are shown in Fig. 5(a) and (b), respectively. MAE values are not plotted since they do not affect the performance, i.e. the MAE is the initial error and it stays constant over the range uncertainties, indicating the good convergence performance of the controller. From these results, we can conclude that the system can perform well even the mass and inertias are either highly over-estimated or underestimated. The RSME stays in the range of about 5.2cm to 6.4cm, while the EPAE stays within 3.2cm.

B. Case II: Center of Mass Location Uncertainties

Another major uncertainty in our model would be the location of the CM. Although the location of the center of mass can deviate also in the y direction in the body fixed robot frame, we assume the robot is reasonably symmetric, hence only the x-directional component is studied. Similar to the rationale noted in the previous section, the RSME and EPAE of x and y of the controller under such uncertainties are shown in Fig. 5(a) and (b), respectively. The MAE values are also not plotted since the controller is able to drive the system to converge to the desired trajectories. From these results, we can also conclude that the system can perform quite well even the existence of uncertainties in center of mass. However, the effect is more dominant than the case in uncertainties in the mass and inertia. Although the RSME is in the range of 4.8cm to 5.7cm (lesser than the previous case), the more important factor of EPAE is in the order of 4.9cm, which is reasonably higher than the previous case.

VII. Conclusion

This paper presented a simple method to control a ubiquitous system of DD-WMR by invoking the DF property

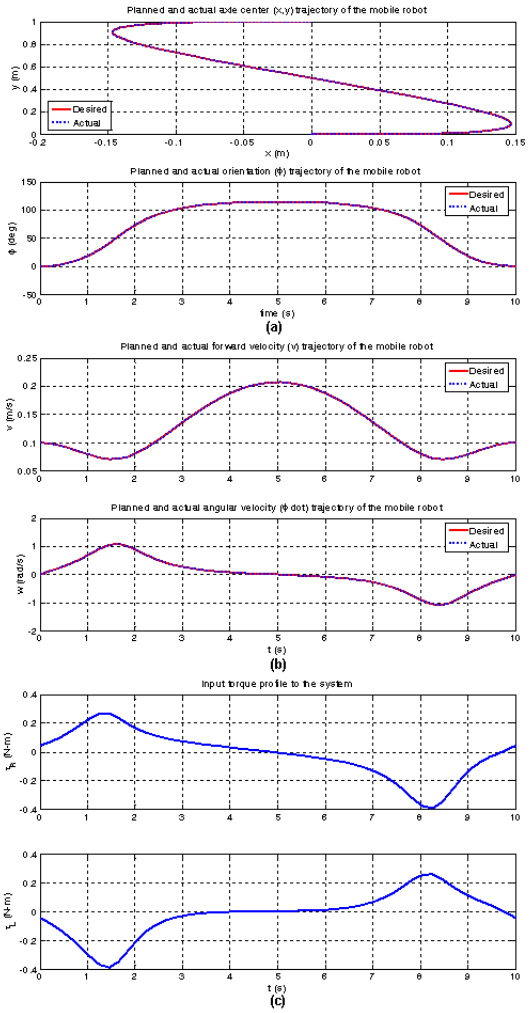

Fig. 2. Case I - "Parallel parking": (a) (x, y, ϕ) position trajectories, (b) (v, ω) velocity trajectories, (c) required left and right wheel torques to accomplish the trajectory.

within the model of the system. We presented the integrated motion planning and control method for both the kinematic and dynamic models. By choosing the same flat outputs and establishing the same diffeomorphism, both the models can be dynamically feedback linearized. It is believed that the method is sufficient to attack large variety of applications. Future work include providing an open-source codes for such implementation online. The development of robust and/or adaptive control schemes to account for model uncertainties are also possible future research avenues.

VIII. Acknowledgments

The author gratefully acknowledge the discussions with V. N. Krovi, S. K. Agrawal, and J.-C. Ryu for this work. The author also gratefully acknowledge the UT Dallas research fund from M. W. Spong for this continuing effort.

References

[1] "iRobot Coporation: Home Page," URL: http://www.irobot.com.

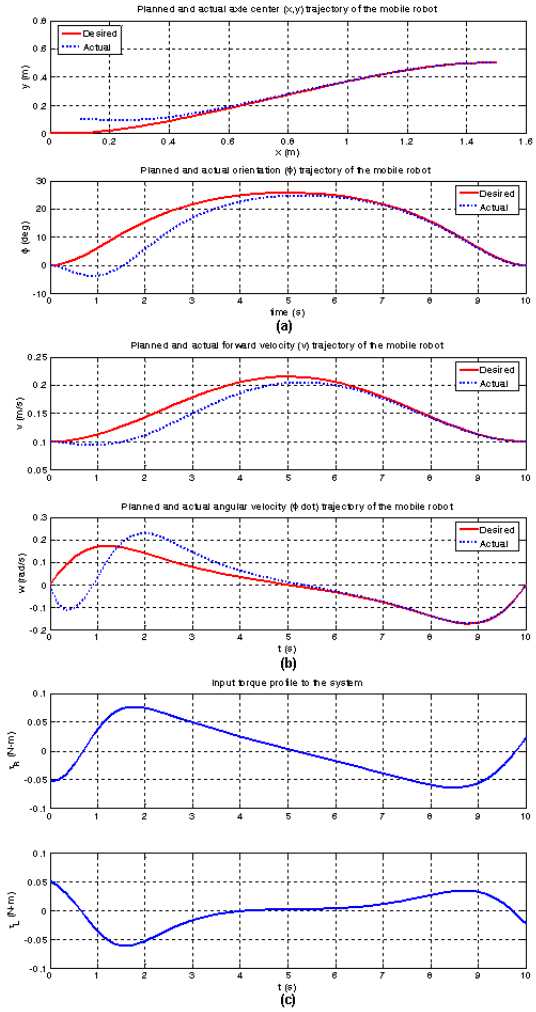

Fig. 3. Case II - Error compensation: (a) (x, y, ϕ) position trajectories, (b) (v, ω) velocity trajectories, (c) required left and right wheel torques to accomplish the trajectory.

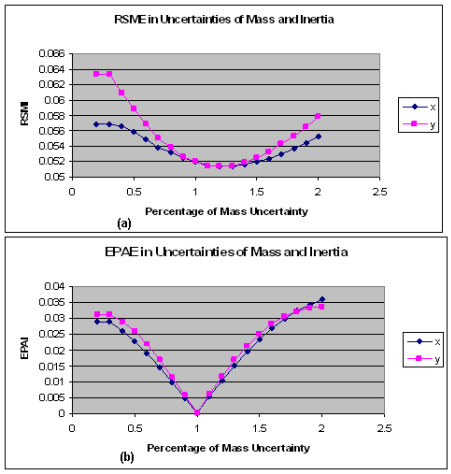

Fig. 4. Case I: Sensitivity to inertial parameters based on (a) RSME and (b) EPAE

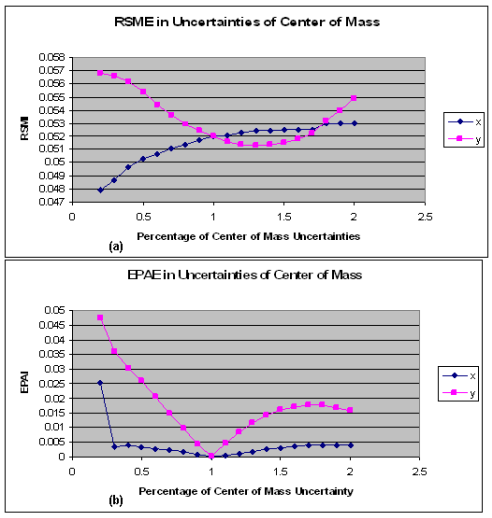

Fig. 5. Case II: Sensitivity to CM location based on (a) RSME and (b) EPAE

TABLE I

TERMINAL CONDITIONS FOR TRAJECTORY PLANNING

	Case I		Case II	
	Initial Conditions	Final Conditions	Initial Conditions	Final Conditions
$x(m)$	0.0	0.0	0.0	1.5
$y(m)$	0.0	1.0	0.0	0.5
$\phi(rad)$	0.0	0.0	0.0	0.0
$v(m/s)$	0.1	0.1	0.1	0.1
$\dot{\phi}(rad/s)$	0.0	0.0	0.0	0.0
$\eta_v(m/s^2)$	0.0	0.0	0.0	0.0

[2] F. Mondada, E. Franzi, and A. Guignard, "The Development of Khepera," in *First International Khepera Workshop*, 1999, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, pp. 7–14.

[3] "AmigoBot Robot for education & collaborative research," URL: http://www.activrobots.com/ROBOTS/amigobot.html.

[4] "Garcia Technology," URL: http://www.acroname.com/technology/104/abstract.html.

[5] "LEGO.com MINDSTORMS NXT," URL: http://mindstorms.lego.com.

[6] "Segway," URL: http://www.segway.com.

[7] "Automated Order Fulfillment Material Handling System - Kiva Systems," URL: http://www.kivasystems.com.

[8] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press, Cambridge, MA, June 2005.

[9] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, Feb 1996.

[10] H. Sira-Ramirez and S. K. Agrawal, *Differential Flat Systems*, Marcel Dekker, New York NY, USA, 2004.

[11] Yun X. Sarkar, N. and V. Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots," *International Journal of Robotics Research*, vol. 13, no. 1, pp. 55–69, February 1994.

[12] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149–163, 1997.

[13] "Real-Time Workshop - Generate C code from Simulink models and MATLAB code," URL: http://www.mathworks.com/products/rtw.