

Article

MPC and PSO Based Control Methodology for Path Tracking of 4WS4WD Vehicles

Qifan Tan ^{1,*} , Penglei Dai ², Zhihao Zhang ³ and Jay Katupitiya ³

¹ School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China

² School of Mechanical and Mechatronic Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia; penglei.dai@uts.edu.au

³ School of Mechanical and Manufacturing Engineering, The University of New South Wales, Sydney, NSW 2052, Australia; zhihao.zhang1@unsw.edu.au (Z.Z.); j.katupitiya@unsw.edu.au (J.K.)

* Correspondence: 14116355@bjtu.edu.cn; Tel.: +86-186-0124-6239

Received: 31 May 2018; Accepted: 13 June 2018; Published: 19 June 2018

Abstract: Four wheel steering and four wheel drive (4WS4WD) vehicles are over-actuated systems with superior performance. Considering the control problem caused by the system nonlinearity and over-actuated characteristics of the 4WS4WD vehicle, this paper presents two methods to enable a 4WS4WD vehicle to accurately follow a predefined path as well as its reference trajectories including velocity and acceleration profiles. The methodologies are based on model predictive control (MPC) and particle swarm optimization (PSO), respectively. The MPC method generates the virtual inputs in the upper controller and then allocates the actual inputs in the lower controller using sequential quadratic programming (SQP), whereas the PSO method is proposed as a fully optimization based method for comparison. Both methods achieve optimization of the steering angles and wheel forces for each of four independent wheels simultaneously in real time. Simulation results achieved by two different controllers in following the reference path with varying disturbances are presented. Discussion about two methodologies is provided based on their theoretical analysis and simulation results.

Keywords: 4WS4WD vehicle; force control; MPC; PSO; path tracking

1. Introduction

With the development of Autonomous Ground Vehicles (AGVs) in the last few decades, the demand for accuracy, maneuverability and controllability in vehicle's navigation is ever increasing. For example, an AGV may be required to follow a path accurately under unstructured and uneven terrain conditions, where a significant amount of wheel slip and unpredictable disturbance forces occur at the vehicle's wheels. The 4WS4WD vehicle, with four wheels that can be steered and driven independently, is a revolutionary platform that has great potential to perform high maneuverability and flexibility in harsh environments.

The main challenge in the control of 4WS4WD control is the number of control inputs (four steering angles and four drive torques), which results in an over-actuated system, where only three outputs including its degree of freedom (DOF) in the longitudinal, lateral and angular directions of the vehicle are concerned. How to allocate all eight control inputs to achieve high path following performance has not yet been effectively solved. The control allocation is proposed to handle the control problems of over-actuated systems [1]. Generally, the control allocation can be treated as an optimization problem.

For controller designing, a model of the vehicle under control is generally required to facilitate the selection of future control inputs. A dynamic model describes the states of the vehicle based on the forces applied. However, the development of a detailed vehicle dynamic model is always a challenging task due to the uncertainty of parameters and the complex disturbances from the external

forces. The majority of existing control methodologies for 4WS4WD vehicles are proposed based on the linearized dynamic model, which lead to the loss of input degree of freedom [2,3]. Meanwhile, some other methods use nonlinear dynamic models in the control of 4WS4WD vehicles [4,5], where control inputs are subjected to some relationship constraints to simplify the controller design. However, in practice, the four independent wheels may interact with different terrain conditions, where different slip, wheel forces and terrain disturbances are generated on the corresponding contact patches. Hence, it is desirable to make four wheels individually controlled, thereby limiting and/or overcoming different slip and disturbance on different contact patch.

There is an abundance of literature that presents kinematic modelling of ground vehicles [6–8], in which the vehicles are assumed to operate at low speeds to reduce the dynamic effects. Most vehicle kinematic models are developed based on the non-integrable kinematic constraints, known as non-holonomic constraints. As a result, the wheel slip has to be ignored with the assumption of zero relative velocity between wheels and terrain [9,10]. To utilize the kinematic model's advantage of keeping the steering control relatively independent of velocity control [11], it is desirable to incorporate wheel slip in the vehicle kinematic model so as to facilitate the accurate vehicle control in complex terrain conditions where the no-slip assumption is not applicable.

To realize force control, a novel type of 4WS4WD vehicle with force sensors at each wheel has been designed as shown in Figure 1. This vehicle possesses the characteristics of independent steering and drive control at each wheel and force measurements. In this work, the dynamic model is partitioned into a hierarchy of three levels to facilitate the incorporation of force sensors and overcome uncertainties in the model. The force sensors at the drive unit allow the measurement of actual force data, while models of the drive unit and tire are used for simulation.

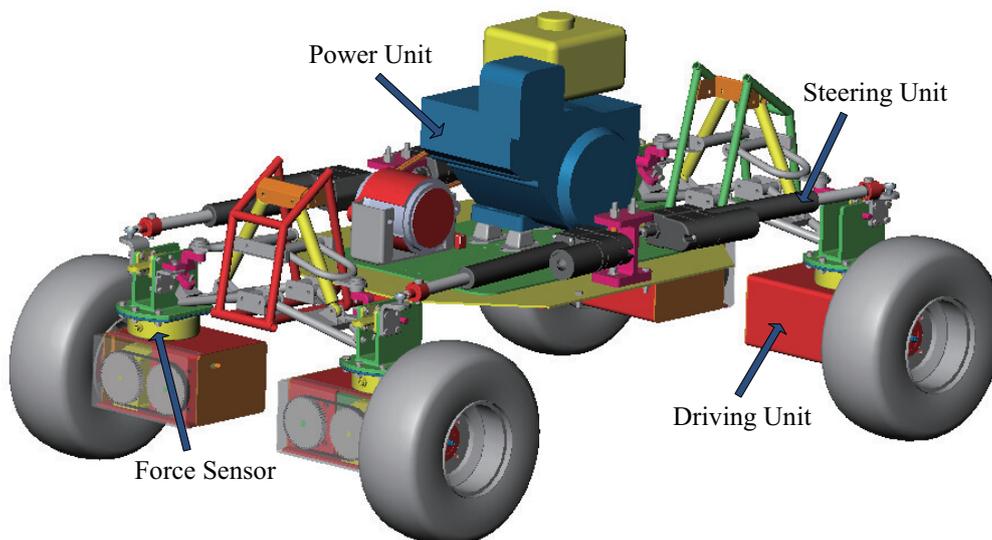


Figure 1. The 4WS4WD (four wheel steering and four wheel drive) vehicle.

Model predictive control (MPC) is selected for its ability in handling linear constraints and time-varying systems as well as its good performance in tracking problems. Particle swarm optimization (PSO) is also selected for its fast searching speed in global optimization. MPC and PSO have been successfully applied in the controller design of real-time control systems [12–17]. In this work, MPC and PSO based control methods are proposed to realize the controlling aim of achieving good path following performance as well as high motion quality via vehicle steering control and independent force control at four wheels.

As novel contributions, the MPC methodology is applied to achieve precise path tracking of 4WS4WD vehicle. Based on the MPC theory, an offline control law is proposed to guarantee the stability of the upper controller. An sequential quadratic programming (SQP) based control allocation

is developed to control the 4WS4WD vehicle in the lower controller. The inclusion of full independent force control and steering control on all four wheels enable the maximization of performance. In this work, comparison of MPC and PSO on the same vehicle model is provided, in which the proposed PSO control methodology is a further refinement of the PSO methodology presented previously in [18]. The PSO control methodology in this paper simplifies the derivation and gives an algorithm in a more general form, which facilitates the comparison with other control methods. In addition, both methodologies are compared with the kinematic model based method proposed in [19].

The paper is organized as follows: Section 2 describes the 4WS4WD vehicle modelling. The MPC and PSO control methodologies are presented in Section 3. In Section 4, simulation setup and the reference path are presented. The result of the two controllers are compared. In Section 5, the discussion about two methodologies are provided based on their theoretical analysis and simulation results. Finally, conclusions are provided in Section 6.

2. Vehicle System Modelling

In order to develop suitable control methodology, the 4WS4WD vehicle must be modelled for controller design and simulation. The model used for this work is a fully dynamic model that consists of three components, vehicle body dynamics to estimate vehicle body movement, drive unit dynamics to model the tire force acting on the drive unit in vehicle coordinate frame and a tire dynamic model to model the force generated by the tire in a wheel coordinate frame. The separation of dynamic model allows force sensors to be incorporated easily and reduces uncertainties by obtaining actual force measurements. In the end, an offset error model is proposed for evaluating the tracking performance and is used in the controller design.

2.1. Vehicle Body Dynamic Model

As shown in Figure 2, the vehicle body dynamic model describes the motion of the vehicle by representing each drive unit as a pair of forces, with F_{Sl} in the longitudinal and F_{SL} in the lateral direction of the corresponding wheel. The reason for separating the dynamics of the vehicle at the drive unit boundary is to allow force sensors to be incorporated to measure the forces acting on each wheel, instead of relying on the tire model to estimate them. To facilitate the notations, the four wheels are numbered by 1, 2, 3 and 4 in circles. In particular, the force acting at each force sensor is decomposed into two components (i.e., F_{Sli} and F_{SLi} , $i = 1, \dots, 4$) normal to each other.

The equation of the vehicle body dynamics can be expressed as

$$acc = \begin{bmatrix} a_r \\ a_l \\ \gamma \end{bmatrix} = M^{-1}C \begin{bmatrix} R_1 F_{AS1} \\ R_2 F_{AS2} \\ R_3 F_{AS3} \\ R_4 F_{AS4} \end{bmatrix}, \tag{1}$$

where the acceleration vector denoted by acc consists of longitudinal acceleration a_l , radial acceleration a_r and angular acceleration γ .

The mass matrix is represented by M :

$$M = \begin{bmatrix} M - 4m_d & 0 & 0 \\ 0 & M - 4m_d & 0 \\ 0 & 0 & J_z \end{bmatrix}, \tag{2}$$

where M is the mass of the vehicle, J_z is the vehicle body inertia, and m_d is the mass of the drive unit.

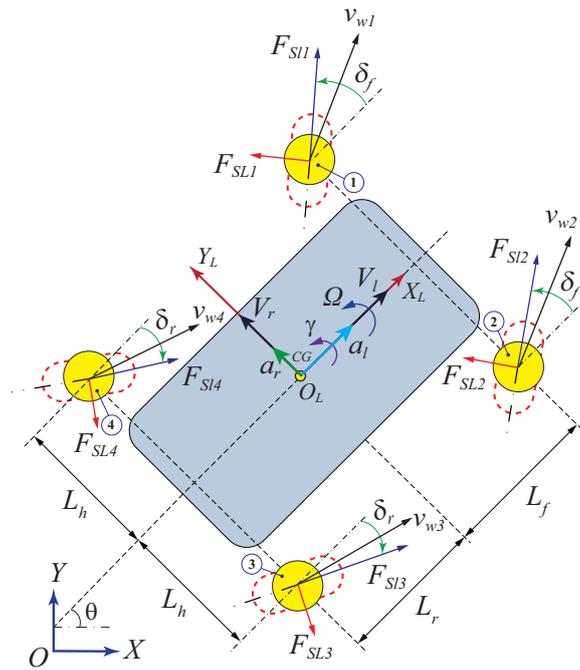


Figure 2. Vehicle body dynamic model.

The matrix C is decided by vehicle dimension, written as

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -L_h & L_f & L_h & L_f & L_h & -L_r & -L_h & -L_r \end{bmatrix}. \quad (3)$$

The force vectors F_{ASi} are defined as

$$F_{ASi} = [F_{SLi} \quad F_{SLi}]^T, \quad i = 1, \dots, 4. \quad (4)$$

The transfer matrices R_i from wheel frame to vehicle local coordinate frame $X_L - O_L - Y_L$ are presented in

$$R_i = \begin{bmatrix} \cos \delta_i & -\sin \delta_i \\ \sin \delta_i & \cos \delta_i \end{bmatrix}, \quad i = 1, \dots, 4. \quad (5)$$

The lateral components F_{SLi} are determined by the lateral forces acting on tires, while the longitudinal components F_{SLi} are viewed as intermediate control inputs of vehicle system, which can be achieved by controlling the torques (i.e., $T_i, i = 1, \dots, 4$) applied on wheels. To simplify the coupled issue of left and right wheel steerings, the steering angles are constrained by

$$\begin{aligned} \delta_1 &= \delta_2 = \delta_f, \\ \delta_3 &= \delta_4 = \delta_r. \end{aligned} \quad (6)$$

Therefore, the vehicle system in this work considers six control inputs (i.e., δ_f, δ_r and $T_i, i = 1, \dots, 4$) in total.

2.2. Driving Unit Dynamic Model

While the force sensors provide measurement of the current forces acting on each wheel, controller design requires prediction of future values based on control inputs. The drive unit dynamic model as shown in Figure 3 is described in this section.

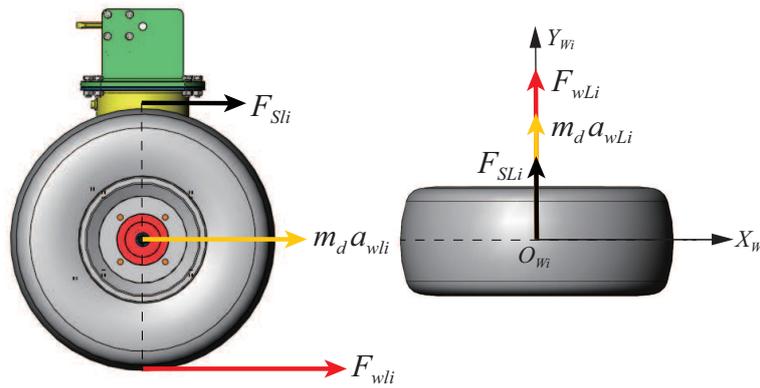


Figure 3. Driving unit dynamic model.

In each driving unit, the force sensor mounted on the wheel hub rotate with the wheel. Force measurement is in the direction of the steering angle. According to Figure 3, the dynamics of driving unit can be expressed by

$$F_{ASi} = [F_{Sli} \quad F_{SLi}]^T = F_{wi} - [m_d a_{wli} \quad m_d a_{wLi}]^T, \tag{7}$$

where a_{wli} and a_{wLi} denote the longitudinal and lateral accelerations of each driving unit i in the wheel coordinate system $(X_{Wi} - O_{Wi} - Y_{Wi})$, respectively. Using the transfer matrices R_i and the accelerations a_l, a_r and γ given in vehicle body dynamic model, a_{wli} and a_{wLi} can be deduced as

$$\begin{aligned} [a_{w1l} \quad a_{w1L}] &= [a_l - L_h \gamma \quad a_r + L_f \gamma] R_1, \\ [a_{w2l} \quad a_{w2L}] &= [a_l + L_h \gamma \quad a_r + L_f \gamma] R_2, \\ [a_{w3l} \quad a_{w3L}] &= [a_l + L_h \gamma \quad a_r - L_r \gamma] R_3, \\ [a_{w4l} \quad a_{w4L}] &= [a_l - L_h \gamma \quad a_r - L_r \gamma] R_4. \end{aligned} \tag{8}$$

In Equation (7), F_{wi} represents the force vector acting on each wheel, which is written as

$$F_{wi} = [F_{wli} \quad F_{wLi}]^T. \tag{9}$$

2.3. Tire Model

The actual generation of wheel forces are from the contact between tires and the ground. The wheel forces depend on the surface friction, load, and the intrinsic properties of the tire. To analyze the wheel forces F_{wi} in Equation (9), the tire model is built as shown in Figure 4.

Considering the generation of wheel forces and external disturbances, F_{wi} can be expressed as

$$F_{wi} = [F_{wli} \quad F_{wLi}]^T = [F_{li} - F_{gli} \quad F_{Li} - F_{gLi}]^T, \tag{10}$$

where F_{li} and F_{Li} ($i = 1, \dots, 4$) are longitudinal and lateral forces caused by wheel slip, respectively. F_{gli} and F_{gLi} denote the terrain disturbances corresponding to F_{li} and F_{Li} .

According to [20], the longitudinal force can be considered proportional to the slip ratio in small range, which is expressed by

$$F_{li} = \begin{cases} k_{li} F_{Ni} \frac{\zeta_i}{0.1}, & |\zeta_i| \leq 0.1, \\ k_{li} F_{Ni}, & |\zeta_i| > 0.1, \end{cases} \tag{11}$$

where F_{Ni} is the weight force on wheel i . In the simulation, the load transfer due to the longitudinal and lateral accelerations and roll angle are taken into consideration. F_{Ni} can be calculated by considering longitudinal and lateral load transfer according to [21]. The longitudinal slip stiffness k_{li} is determined by the tire type and terrain condition. ζ_i is the longitudinal slip ratio of wheel i presented in [22]:

$$\zeta_i = \begin{cases} \frac{R_w\omega_i - v_{wi}}{R_w\omega_i}, & R_w\omega_i > v_{wi} > 0, \\ \frac{R_w\omega_i - v_{wi}}{v_{wi}}, & R_wv_{wi} > \omega_i > 0, \\ 0, & R_wv_{wi} = \omega_i \geq 0, \end{cases} \quad (12)$$

where R_w and ω_i denote the tire radius and angular velocity of wheel i as shown in Figure 4b. v_{wi} represents the actual velocity of the wheel i , which can be calculated by the vehicle geometry as well as velocities V_L , V_r and Ω shown in Figure 2. Note that the model is proposed only considering the vehicle is moving forward. Thus, v_{wi} and ω_i are set to be nonnegative. As is discussed in Equation (12), the slip ratio is zero when the vehicle is resting.

Finally, as per Figure 4b, the wheel dynamic equation indicating the relationship between T_i and F_{ji} can be written as

$$T_i = (F_{li} + F_{ri})R_w + \dot{\omega}_i J_w, \quad (13)$$

where J_w represents the wheel inertia, and F_{ri} is the rolling resistance calculated by

$$F_{ri} = F_{Ni}(K_{r0} + K_{r1}V_L^2), \quad (14)$$

where $K_{r0} = 0.015$ and $K_{r1} = 7 \times 10^{-6} \text{ s}^2/\text{m}^2$ are the parameters for common car tires [23].

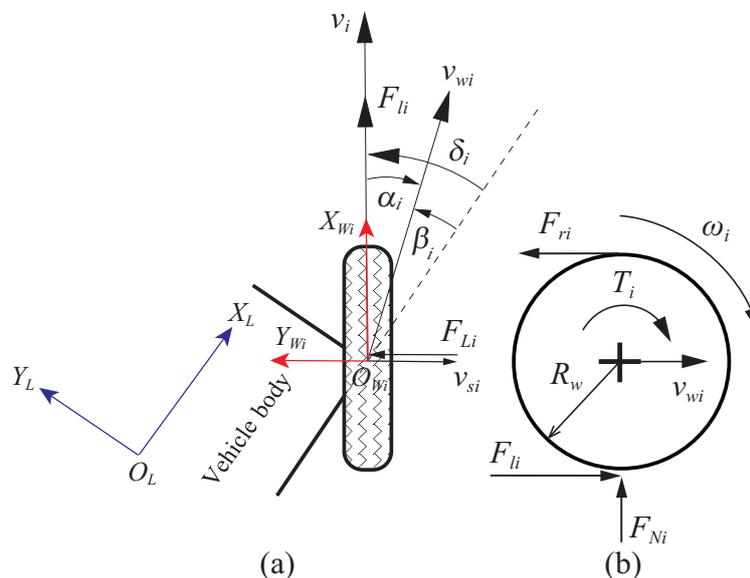


Figure 4. Tire model. (a) Top view of the tire; (b) lateral view of the tire.

The steering motion of wheels result in the lateral slip velocities v_{si} , which causes the actual direction of wheel velocity v_{wi} to differ from the wheel center plane by the slip angle α_i . According to the tire lateral characteristic curve [24], a linear model for F_{Li} is given in

$$F_{Li} = \begin{cases} -k_{Li}F_{Ni} \frac{\alpha_i}{5}, & |\alpha_i| \leq 5^\circ, \\ -k_{Li}F_{Ni}, & \alpha_i > 5^\circ, \\ k_{Li}F_{Ni}, & \alpha_i < -5^\circ, \end{cases} \quad (15)$$

where k_{Li} is the lateral slip stiffness of the wheel i . Note that, due to the sign conventions in $X_{Wi} - O_{Wi} - Y_{Wi}$ and $X_L - O_L - Y_L$, a negative slip angle causes a positive lateral force and vice versa.

As shown in Figure 4a, the slip angle α_i can be calculated by steering angle δ_i and side slip angle β_i ,

$$\alpha_i = \beta_i - \delta_i, \quad (16)$$

where the expression of β_i is available in our previous work presented in [25].

2.4. Offset Model

An offset error model is proposed for the evaluation of the performance of the controllers. The reference position (RP) of the vehicle on the reference path is defined as the normal projection of the heading of the vehicle at the centre of gravity on to the reference path, as shown in Figure 5. The tracking error in vehicle coordinate frame consists of lateral offset error l_{os} and heading error θ_{os} , expressed as

$$\widetilde{pos} = [0 \quad l_{os} \quad \theta_{os}]^T \quad (17)$$

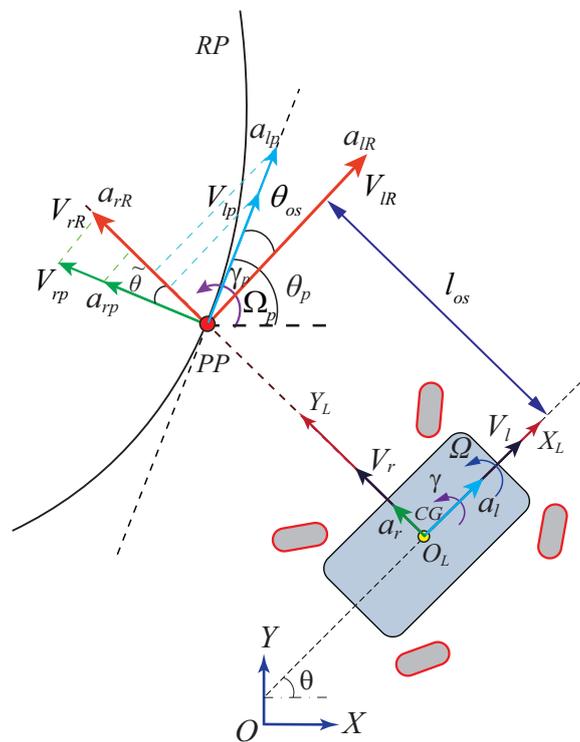


Figure 5. Offset model.

The position errors are always in the lateral direction of the vehicle. The longitudinal position error is set to 0. The heading error can be found by

$$\theta_{os} = \theta - \theta_{ref}, \quad (18)$$

where θ_{ref} is the reference heading direction of the RP, tangential to the reference path.

The velocity error states \widetilde{vel} is the difference between vehicle velocity vel and reference velocity states vel_{ref} in the vehicle coordinate frame. \widetilde{vel} can be expressed as

$$\widetilde{vel} = \begin{bmatrix} V_{lerr} & V_{rerr} & \Omega_{err} \end{bmatrix}^T = vel - vel_{ref}, \tag{19}$$

where V_{lerr} , V_{rerr} and Ω_{err} are the errors in longitudinal, lateral, and angular velocity, respectively. Reference velocity states are defined as

$$vel_{ref} = \begin{bmatrix} V_{lref} & V_{rref} & \Omega_{ref} \end{bmatrix}^T, \tag{20}$$

where reference angular velocity Ω_{ref} is the rate of change of the heading of RP on the reference path.

The reference longitudinal velocity V_{lref} and lateral velocity V_{rref} in vehicle coordinate frames can be derived from the reference path coordinate frame by the equations:

$$\begin{aligned} V_{lref} &= V_{lp} \cos(\theta_{os}) - V_{rp} \sin(\theta_{os}), \\ V_{rref} &= V_{lp} \sin(\theta_{os}) + V_{rp} \cos(\theta_{os}), \end{aligned} \tag{21}$$

where V_{lp} is the reference longitudinal velocity, taken as tangential to the reference path. V_{rp} is the reference radial velocity normal to the reference path at RP.

Finally, the acceleration error states \widetilde{acc} are written as:

$$\widetilde{acc} = \begin{bmatrix} a_{lerr} & a_{rerr} & \gamma_{err} \end{bmatrix}^T = acc - acc_{ref}, \tag{22}$$

where acc is the difference between vehicle acceleration acc and reference acceleration acc_{ref} . The reference acceleration vector acc_{ref} is defined as

$$acc_{ref} = \begin{bmatrix} a_{lref} & a_{rref} & \gamma_{ref} \end{bmatrix}^T, \tag{23}$$

where a_{lref} and a_{rref} are, respectively, the reference longitudinal and lateral acceleration at point RP, measured in the direction and normal to the vehicle heading, and γ_{ref} is the reference angular acceleration of the point RP.

The reference acceleration in vehicle coordinate frame can be derived from the path coordinate frame by

$$\begin{aligned} a_{lref} &= a_{lp} \cos(\theta_{os}) - a_{rp} \sin(\theta_{os}), \\ a_{rref} &= a_{lp} \sin(\theta_{os}) + a_{rp} \cos(\theta_{os}), \end{aligned} \tag{24}$$

where a_{lp} and a_{rp} are the reference acceleration tangential and normal to the reference path.

3. Control Methodology

Two methods of finding the optimal control inputs for steering angles (i.e., δ_f and δ_r) and drive forces F_{Sli} , ($i = 1, \dots, 4$) are developed. The first method used model predictive control (MPC) with a sequential quadratic programming (SQP) solver. The second method is particle swarm optimization (PSO). The objective of the controller is to accurately follow a predetermined path through multiple terrain types.

3.1. MPC-SQP Method

3.1.1. Offset Model Linearization

In order to be applied in the MPC controller design, the offset model is linearized by a feedback linearization method. By defining the state vector $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$, as $x_1 = V_{lerr}$, $x_2 = l_{os}$, $x_3 = \dot{l}_{os}$, $x_4 = \theta_{os}$ and $x_5 = \dot{\theta}_{os}$, the offset model can be written in the following form:

$$\begin{aligned} \dot{x}_1 &= a_r - (a_{lp} \cos x_4 - a_{rp} \sin x_4), \\ \dot{x}_2 &= x_3, \\ \dot{x}_3 &= a_r - (a_{lp} \sin x_4 + a_{rp} \cos x_4), \\ \dot{x}_4 &= x_5, \\ \dot{x}_5 &= \gamma - \gamma_{ref}. \end{aligned} \tag{25}$$

In this model, a_r , a_l and γ are considered as the inputs while other constants including reference accelerations and velocities can be obtained from the vehicle states. The nonlinearity of the model comes from trigonometric terms of x_4 . For accurate path tracking problems, the yaw error is assumed to vary smoothly within $[-10^\circ 10^\circ]$. Then, the model can be linearized by feedback linearization. Define the input vector $\mathbf{u} = [a_r \ a_l \ \gamma]^T$; then, the new input vector at time t_k is obtained as

$$\mathbf{v}(\mathbf{x}, t)|_{t=t_k} = \mathbf{u} - \begin{bmatrix} a_{lp} \cos x_4 - a_{rp} \sin x_4 \\ a_{lp} \sin x_4 + a_{rp} \cos x_4 \\ \gamma_{ref} \end{bmatrix}_{x_4=x_4(t_k)}, \tag{26}$$

and the offset model is expressed as

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{v}(\mathbf{x}, t)|_{t=t_k}, \tag{27}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It can be seen that the model is time-varying but can be treated as a linear model at each sampling step. The output vector is denoted by \mathbf{y}_c and the equation is written as

$$\mathbf{y}_c = \mathbf{C}_c \mathbf{x}, \tag{28}$$

where

$$\mathbf{C}_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

3.1.2. Model Predictive Control

To develop the MPC controller, the offset model as given in Equations (27) and (28) needs to be discretized and its discrete-time state vector form can be written as

$$\begin{aligned} \mathbf{x}_d(k+1) &= \mathbf{A}_d \mathbf{x}_d(k) + \mathbf{B}_d \mathbf{u}_d(k), \\ \mathbf{y}_d(k) &= \mathbf{C}_d \mathbf{x}_d(k), \end{aligned} \tag{29}$$

where $\mathbf{x}_d(k)$, $\mathbf{y}_d(k)$ and $\mathbf{u}_d(k)$ are the state vector, output vector and input vector, respectively. The coefficient matrices \mathbf{A}_d , \mathbf{B}_d and \mathbf{C}_d are updated after discretization.

To eliminate undesirable oscillations, embedded integrator vectors $\Delta \mathbf{x}_d(k) = \mathbf{x}_d(k+1) - \mathbf{x}_d(k)$, $\Delta \mathbf{u}(k) = \mathbf{u}_d(k+1) - \mathbf{u}_d(k)$ are defined, thereby an augmented state-space model can be expressed as

$$\begin{aligned} \begin{bmatrix} \Delta \mathbf{x}_d(k+1) \\ \mathbf{y}_d(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_d & \mathbf{O}^T \\ \mathbf{C}_d \mathbf{A}_d & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_d(k) \\ \mathbf{y}_d(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_d \\ \mathbf{C}_d \mathbf{B}_d \end{bmatrix} \Delta \mathbf{u}(k), \\ \mathbf{y}_d(k) &= \begin{bmatrix} \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_d(k) \\ \mathbf{y}_d(k) \end{bmatrix}, \end{aligned} \tag{30}$$

where \mathbf{O} is a zero matrix, and \mathbf{I} represents the identity matrix.

Defining the new state vector $\mathbf{x}(k) = [\Delta \mathbf{x}_d(k)^T \ \mathbf{y}_d(k)^T]^T$, the augmented model can be written in the following matrix form:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} \Delta \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C} \mathbf{x}(k), \end{aligned} \tag{31}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{O}^T \\ \mathbf{C}_d \mathbf{A}_d & \mathbf{I} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_d \\ \mathbf{C}_d \mathbf{B}_d \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{O}^T & \mathbf{I} \end{bmatrix}.$$

Theorem 1. *Given a discrete time system following the form of Equation (31), the asymptotic stabilization of the closed-loop system can be realized by substituting the first item of $\Delta \mathbf{U}^*$ as the control input $\Delta \mathbf{u}(k)$, when $\Delta \mathbf{U}^*$ is the optimal solution of the following optimization problem:*

$$\begin{aligned} \arg \min_{\Delta \mathbf{U}} J &= (\mathbf{R}_s - \mathbf{Y})^T (\mathbf{R}_s - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{\mathbf{R}} \Delta \mathbf{U}, \\ \text{s.t.} \quad y_{N_p}(\Delta \mathbf{U}) &= 0, \end{aligned} \tag{32}$$

where \mathbf{Y} denotes the predicted output sequence, $\Delta \mathbf{U}$ denotes the future input sequence, and N_p is the prediction horizon. \mathbf{R}_s is the sequence of the control target vector. $y_{N_p}(\Delta \mathbf{U})$ represents the error between the final predicted output and target.

Proof of Theorem 1. To prove the stability, the Lyapunov function $V(x_k)$ is defined equal to the value of the objective function J_k subjected to its optimal solution, which can be expressed as

$$\begin{aligned} V(x_k) &= \min J_k \\ &= \sum_{i=1}^{N_p} \mathbf{y}_{k+i}^T \mathbf{y}_{k+i} + \sum_{i=0}^{N_p-1} \Delta \mathbf{u}_{k+i}^T r_w \Delta \mathbf{u}_{k+i}, \end{aligned} \tag{33}$$

where, $\Delta u_k, \dots, \Delta u_{k+N_p-1}$ are obtained by the optimal solution of future inputs, and y_k, \dots, y_{k+N_p} represent the corresponding error sequence between future outputs and target. r_w is the nonnegative gain matrix.

According to the definition in Equation (33), the Lyapunov function at the next sample $k + 1$ is written as

$$V(x_{k+1}) = \sum_{i=1}^{N_p} y_{k+1+i}^T y_{k+1+i} + \sum_{i=0}^{N_p-1} \Delta u_{k+1+i}^T r_w \Delta u_{k+1+i}. \tag{34}$$

To facilitate the comparison between two neighboring Lyapunov function values, a intermediate function \bar{V} is defined, which is formed by evaluating $V(x_{k+1})$ with a defined inputs sequence, which is obtained by shifting the optimal inputs sequence of $V(x_{ki})$ one step forward, and setting its last input Δu_{k+N_p} as zero. It is obvious that the objective function value of non-optimal inputs sequence has to be no less than $V(x_{ki+1})$, which can be expressed as

$$V(x_{k+1}) \leq \bar{V}, \tag{35}$$

thereby,

$$V(x_{k+1}) - V(x_k) \leq \bar{V} - V(x_k). \tag{36}$$

Since \bar{V} shares the same future inputs sequence and the predictive outputs sequence with $V(x_k)$ for the sample time $k + 1, \dots, k + N_p - 1$, it can be easily derived that the difference between these two functions is

$$\bar{V} - V(x_k) = y_{k+N_p}^T y_{k+N_p} - y_{k+1}^T y_{k+1} - \Delta u_k^T r_w \Delta u_k. \tag{37}$$

As is given in Equation (32), the optimization problem is subjected to the constraint $y_{k+N_p} = 0$. Then, it can be obtained that

$$\bar{V} - V(x_k) \leq -y_{k+1}^T y_{k+1} - \Delta u_k^T r_w \Delta u_k. \tag{38}$$

Then, the monotonicity of the Lyapunov function can be obtained by

$$V(x_{k+1}) - V(x_k) \leq -y_{k+1}^T y_{k+1} - \Delta u_k^T r_w \Delta u_k < 0, \tag{39}$$

which can prove the asymptotic stability of the system. \square

The model predictive control algorithm is realized by receding optimization. In order to apply the MPC efficiently, we assume that the predicted outputs sequence is in a finite prediction horizon N_p and the inputs sequence is in a control horizon N_c , which is less than N_p . The sequences mentioned above can be expressed in the matrix form:

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} y(k+1|k) & y(k+2|k) & \dots & y(k+N_p|k) \end{bmatrix}^T, \\ \Delta \mathbf{U} &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_c-1) \end{bmatrix}^T, \end{aligned} \tag{40}$$

where $y(k+n|k)$ denotes the predicted outputs at time $k+n$ based on the states at time k . Based on Theorem 1 and the assumption above, the corollary about finite-time unconstrained MPC can be obtained as follows.

Corollary 1. *Given the system without input and output constraints, the prediction and control horizon are N_p and N_c , respectively. Then, the following feedback control law $\Delta u(k) = Kx(k)$ can asymptotically stabilize the closed-loop system, where*

$$K = \overbrace{[1 \ 0 \ \dots \ 0]^T}^{N_c} (-\Xi[\Gamma - (\Psi\Xi)^{-1}(CA^{N_p} + \Psi\Xi\Gamma)]), \tag{41}$$

and,

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \quad \Psi = \begin{bmatrix} CA^{N_p-1}B \\ CA^{N_p-2}B \\ \vdots \\ CA^{N_p-N_c}B \end{bmatrix}^T, \tag{42}$$

$$\Phi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ CA^2B & CAB & \dots & 0 \\ \vdots & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & \dots & CA^{N_p-N_c}B \end{bmatrix},$$

$$\Xi = (\Phi^T\Phi + \bar{R})^{-1},$$

$$\Gamma = \Phi^TF.$$

Proof of Corollary 1. According to Equations (31) and (40), the predicted output sequence Y can be expressed by

$$Y = Fx(k) + \Phi\Delta U. \tag{43}$$

Meanwhile, the final item of Y can be expressed as

$$y(k + N_p|k) = CA^{N_p}x(k) + \Psi\Delta U. \tag{44}$$

By substituting Equations (43) and (44) into Theorem 1, the optimization problem turns into the following form:

$$\begin{aligned} \arg \min_{\Delta U} J &= (R_s - Fx(k_i) - \Phi\Delta U)^T (R_s - Fx(k_i) - \Phi\Delta U) \\ &\quad + \Delta U^T \bar{R} \Delta U, \\ \text{s.t.} \quad \Psi\Delta U + CA^{N_p}x(k_i) &= 0, \end{aligned} \tag{45}$$

where \bar{R} is a weighting matrix.

Note that in the application of offset model based path tracking, the target vector R_s should be zero all the time. It can be seen that J meets an equality constrained quadratic programming. Then, the objective function is expanded by Lagrange expression and simplified by omitting the constant term,

$$J = 2\Delta U^T \Phi^T Fx(k) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U + \zeta^T (\Psi\Delta U + CA^{N_p}x(k)), \tag{46}$$

where ζ is the Lagrange multiplier.

According to the Lagrange multiplier method, the optimal control input vector ΔU^* can be found by solving the following equation system. The solution is obtained by taking the first partial derivatives of J with respect to the vectors ΔU and λ , and then equating these derivatives to zero:

$$\begin{aligned} \frac{\partial J}{\partial \Delta U} &= \Delta U^T (\Phi^T \Phi + \bar{R}) + \Phi^T Fx(k) + \Psi^T \zeta = 0, \\ \frac{\partial J}{\partial \zeta} &= \Psi\Delta U + CA^{N_p}x(k) = 0. \end{aligned} \tag{47}$$

Then, its optimal solution can be obtained:

$$\Delta \mathbf{U}^* = -\mathbf{\Xi}[\mathbf{\Gamma} - (\mathbf{\Psi}\mathbf{\Xi})^{-1}(\mathbf{C}A^{N_p} + \mathbf{\Psi}\mathbf{\Xi}\mathbf{\Gamma})]x(k), \tag{48}$$

where

$$\begin{aligned} \mathbf{\Xi} &= (\mathbf{\Phi}^T\mathbf{\Phi} + \bar{\mathbf{R}})^{-1}, \\ \mathbf{\Gamma} &= \mathbf{\Phi}^T\mathbf{F}. \end{aligned}$$

According to the receding horizon control principle, the first increment of $\Delta \mathbf{U}^*$ is applied as the control inputs. Then, the control law in Equation (41) can be obtained. \square

It can be seen that Corollary 1 gives an offline solution of the MPC algorithm, which significantly improves its computing efficiency. Based on Corollary 1, the desired accelerations $\tilde{\mathbf{A}} = [\tilde{a}_r, \tilde{a}_l, \tilde{\gamma}]^T$ can be obtained by integrating the optimal solution $\Delta u(k)^*$.

3.1.3. Sequential Quadratic Programming Based Control Allocation

Considering the vehicle body with the mass M and inertia J in this work, the command force vector $\boldsymbol{\tau}$ is defined as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_l \\ \tau_r \\ \tau_\gamma \end{bmatrix} = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & J \end{bmatrix} \tilde{\mathbf{A}}, \tag{49}$$

where τ_l , τ_r and τ_γ are the longitudinal and lateral forces and the moment about a vertical axis of the vehicle, respectively.

According to the vehicle body dynamic model given in Equation (1), the actual actuating forces come from the longitudinal forces F_{Sl} and lateral forces F_{SL} on each wheel. Let the command forces $\boldsymbol{\tau}$ produced jointly by the wheel forces and steering angles be expressed as

$$\boldsymbol{\tau}(F_{Sl}, F_{SL}, \delta) = \mathbf{B}_u(\delta)F_{Sl} + \mathbf{B}_w(\delta)F_{SL}, \tag{50}$$

where the i -th column of $\mathbf{B}_u(\delta)$ and $\mathbf{B}_w(\delta)$ can be written as

$$\mathbf{B}_u^i(u_{\delta_i}) = \begin{bmatrix} \cos \delta_i \\ \sin \delta_i \\ -L_{yi} \cos \delta_i + L_{xi} \sin \delta_i \end{bmatrix}, \quad \mathbf{B}_w^i(u_{\delta_i}) = \begin{bmatrix} -\sin \delta_i \\ \cos \delta_i \\ L_{yi} \sin \delta_i + L_{xi} \cos \delta_i \end{bmatrix}.$$

(L_{xi}, L_{yi}) represents the location of each wheel in a coordinate system with its origin at the centre of gravity and positive x -axis forward.

On a 4WS4WD vehicle, the drive forces F_{Sl} and the steering angles δ are the direct inputs while the lateral forces F_{SL} obtained from sensors are considered as a measured disturbance. In this work, the steering angles δ are composed of δ_f and δ_r , which represent the front and rear steering inputs, respectively. Thus, the control problem is reduced to obtaining the feasible solution of Equation (50). In order to facilitate the computation, a slack vector \mathbf{s} is defined by

$$\mathbf{s} = \boldsymbol{\tau} - \mathbf{B}_u(\delta)F_{Sl} + \mathbf{B}_w(\delta)\hat{F}_{SL}, \tag{51}$$

which denotes the error between the commanded and actual generalized forces. The slack variable \mathbf{s} guarantees that there always exists a feasible solution in the following optimization [26].

In order to solve this control allocation problem, the objective function is defined with respect to F_{Sl} , δ and s ,

$$\begin{aligned}
 J(F_{Sl}, \delta, s) &= F_{Sl}^T Q_f F_{Sl} + (\delta - \delta^*)^T Q_\delta (\delta - \delta^*) + s^T Q_s s, \\
 \text{s.t.} \quad s &= \tau - B_u(\delta) F_{Sl} + B_w(\delta) \hat{F}_{Sl}, \\
 s &\in \mathfrak{B}_s, \\
 F_{Sl} &\in \mathfrak{B}_{F_{Sl}}, \\
 \delta &\in \mathfrak{B}_\delta,
 \end{aligned} \tag{52}$$

where \mathfrak{B}_s , $\mathfrak{B}_{F_{Sl}}$ and \mathfrak{B}_δ are the search bounds of each variable.

In this function, the first term minimizes the magnitudes of the drive forces; the second term is used to ensure the steering angle to search around its previous value. By penalizing the slack variable s in the third term, the actual generalized force vector coincides as much as possible with the commanded forces τ . The matrix $Q_f \in I^{4 \times 4}$, $Q_\delta \in I^{2 \times 2}$ and $Q_s \in I^{3 \times 3}$ are used to tune the objective. The search bounds of all variables (i.e. F_{Sl} , δ and s) are specified by the constraints.

Based on the objective function presented above, the control allocation is converted to a nonlinear constrained optimization problem. Using the sequential quadratic programming (SQP), the optimal solution can be computed efficiently and reliably by standard numerical software.

3.2. PSO-Based Method

For the PSO-based method, an objective function including vehicle error states is proposed using the sliding surfaces. In Sliding Mode Control (SMC), the time-varying sliding surface is normally defined by the scalar equation $s(x; t) = 0$, in which $s(x; t)$ is expressed by [27],

$$s(x; t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \tilde{x}, \tag{53}$$

where λ is a positive constant and \tilde{x} is the error state vector.

According to the idea of SMC, the problem of maintaining $\tilde{x} = 0$ is transformed into keeping $s = 0$. In this work, the scalar quantities composed of vehicle error states are introduced into the definition of objective function. Instead of designing the switching control law in SMC, the optimization is used to maintain the scalar quantities on the sliding surface $s = 0$. The vehicle error state vectors including \widetilde{pos} , \widetilde{vel} and \widetilde{acc} are following the definitions in the offset model. Therefore, to track the trajectories of the RP, the vector $s = [s_l, s_r, s_a]$ needs to be defined correspondingly. Note that vector s follows a different definition than that in SQP. It is defined to facilitate the notations in the local boundary part.

For \widetilde{pos} presented in Equation (17), its first component, which represents the position error in the longitudinal direction, is always zero. Therefore, according to Equation (53), the longitudinal scalar quantity s_l can be obtained as

$$s_l = a_{lerr} + \lambda_l V_{lerr}, \tag{54}$$

where a_{lerr} and V_{lerr} are the longitudinal components of \widetilde{acc} and \widetilde{vel} , which are given in Equations (22) and (19), respectively.

To maintain the vehicle on the RP geometrically, another first order scalar quantity s_r is designed as

$$s_r = V_{rerr} + \lambda_r l_{os}, \tag{55}$$

which aims to minimize the offset errors l_{os} and V_{rerr} .

The angular acceleration error $\tilde{\gamma}$ is included to consider the effects of forces and yaw movements of vehicle, and thus a second order scalar quantity is chosen as

$$s_a = \gamma_{err} + 2\lambda_a \Omega_{err} + \lambda_a^2 \theta_{os}, \tag{56}$$

where θ_{os} , Ω_{err} and γ_{err} are given in Equations (17), (19) and (22), respectively.

Then, the problem of following the trajectories of RP is transformed into maintaining s at 0. Using the linear scalarization, an objective function is defined as

$$J_{\min}(F_{Sl}, \delta) = C_l |s_l| + C_r |s_r| + C_a |s_a|, \tag{57}$$

where C_l, C_r and C_a are the weighting coefficients strictly positive and constrained by

$$C_l + C_r + C_a = 1. \tag{58}$$

The variables of objective function in Equation (57) consist of the forces F_{Sli} , steering angles δ_f and δ_r , which can be written as a variable vector,

$$v_{obj} = [\delta_f \quad \delta_r \quad F_{Sl1} \quad F_{Sl2} \quad F_{Sl3} \quad F_{Sl4}]^T \tag{59}$$

First invented by Kennedy and Eberhart (1995), PSO has been successfully applied to solve problems featuring nonlinearity, non-differentiable, and multiple optima. PSO is found to be capable of generating high quality solutions with more stable and faster convergence characteristics, and shorter calculation times than other stochastic methods [17]. For standard PSO at time t , the updating velocity $v_i(t)$ and position $x_i(t)$ of the i -th particle are presented in the following equations:

$$\begin{aligned} v_i(t+1) &= \zeta v_i(t) + \phi_1 \eta_1 (p_b - x_i(t)) + \phi_2 \eta_2 (g_b - x_i(t)), \\ x_i(t+1) &= x_i(t) + v_i(t+1), \end{aligned} \tag{60}$$

where $v_i(t)$ and $x_i(t)$ are vectors in multi-dimensional space. p_b and g_b denote the local optimal position and the global optimal position, respectively. The particle inertia weight is represented by ζ . The particle cognitive acceleration and social acceleration are denoted by ϕ_1 and ϕ_2 , which are defined as positive constants. η_1 and η_2 are two stochastic parameters within [0 1].

The search space of PSO in this work is defined as a six-dimensional space corresponding to the dimension of v_{obj} . Therefore, the particle position vector $x_i(i)$ in Equation (60) represents a possible solution of the objective function in Equation (57).

3.3. Boundary Definition

Given that both methods are reduced to the optimization problems, the definition of the search space determines the quality of the solution.

3.3.1. Global Boundaries

In this work, the global boundaries can be assigned based on the properties of each actuator, which is defined as

$$\mathfrak{B}_g = \begin{cases} \delta_{\min} \leq \delta_f \leq \delta_{\max}, \\ \delta_{\min} \leq \delta_r \leq \delta_{\max}, \\ F_{d\min} \leq F_{Sli} \leq F_{d\max}, \end{cases} \tag{61}$$

where δ_{\min} and δ_{\max} are the minimum and maximum steering angles of each wheel. $F_{d\min}$ and $F_{d\max}$ are the minimum and maximum drive forces provided by the driving unit, which can be calculated by

$$F_{d\min} = -\frac{T_{\max}}{R_w}, \quad F_{d\max} = \frac{T_{\max}}{R_w}, \tag{62}$$

where T_{\max} is the maximum torque that can be achieved by the driving unit of the vehicle.

3.3.2. Local Boundaries

To realize real-time optimization, the computing time obtaining optimal values for variables $[F_{Sli}, \delta]$ needs to be constrained within the sample time of controller T_s . According to the properties of each actuator, the maximum variations of $[F_{Sli}, \delta]$ within T_s can be obtained. In this work, to improve the computing efficiency, the local boundaries are also determined by the states of s .

According to a_r in Equation (1), when $s_l < 0$, the forces F_{Sli} need to be increased, thereby dragging s_l in Equation (54) towards the surface $s_l = 0$. Similarly, when $s_l > 0$, the forces F_{Sli} need to be decreased. Therefore, the local boundaries of F_{Sli} can be written as

$$\mathfrak{B}_{F_{Sli}}^P = \begin{cases} F_{Sli}(t) \leq F_{Sli}(t + T_s) \leq F_{Sli}(t) + \Delta F_{d\max}, & s_l < 0, \\ F_{Sli}(t) - \Delta F_{d\max} \leq F_{Sli}(t + T_s) \leq F_{Sli}(t), & s_l \geq 0, \end{cases} \quad (63)$$

where $\Delta F_{d\max}$ is the maximum change of F_{Sli} that can be achieved within T_s . At the time $t + T_s$, $F_{Sli}(t + T_s)$ are the possible optimal solutions.

For the steering angles δ_f and δ_r in v_{obj} , they affect the vehicle lateral and angular motions in a coupled way. To solve this problem, an allocating method is specified that the vehicle lateral error determines searching direction of δ_f , while δ_r is related to vehicle angular error. When $s_r < 0$, δ_f needs to be increased to drive s_r back to the surface $s_r = 0$, and vice versa. Thus, the local boundary of δ_f is summarized as,

$$\mathfrak{B}_{\delta_f}^P = \begin{cases} \delta_f(t) \leq \delta_f(t + T_s) \leq \delta_f(t) + \Delta\delta_{\max}, & s_r < 0, \\ \delta_f(t) - \Delta\delta_{\max} \leq \delta_f(t + T_s) \leq \delta_f(t), & s_r \geq 0, \end{cases} \quad (64)$$

where $\Delta\delta_{\max}$ is the maximum change of steering angle that can be achieved within T_s . At the time $t + T_s$, the possible optimal solution of front steering angle is $\delta_f(t + T_s)$.

Similarly, based on s_r , the local boundary of δ_r is defined as

$$\mathfrak{B}_{\delta_r}^P = \begin{cases} \delta_r(t) - \Delta\delta_{\max} \leq \delta_r(t + T_s) \leq \delta_r(t), & s_a < 0, \\ \delta_r(t) \leq \delta_r(t + T_s) \leq \delta_r(t) + \Delta\delta_{\max}, & s_a \geq 0. \end{cases} \quad (65)$$

For the PSO in particular, its particle velocity boundaries define the range of speed that particles can achieve to search for the optimal solution. To improve search performance in PSO, the absolute maximum particle velocity is normally set as a certain percentage of particle position range [28]. According to Equations (63)–(65), the particle velocity boundaries can be obtained as

$$\begin{aligned} \mathfrak{B}_{F_{Sli}}^V &= \sigma_F \begin{bmatrix} -\Delta F_{d\max} & \Delta F_{d\max} \end{bmatrix} \\ \mathfrak{B}_{\delta_f}^V &= \sigma_{\delta_f} \begin{bmatrix} -\Delta\delta_{\max} & \Delta\delta_{\max} \end{bmatrix} \\ \mathfrak{B}_{\delta_r}^V &= \sigma_{\delta_r} \begin{bmatrix} -\Delta\delta_{\max} & \Delta\delta_{\max} \end{bmatrix} \end{aligned} \quad (66)$$

where σ_F , σ_{δ_f} and σ_{δ_r} are the particle velocity coefficients for F_{Sli} , δ_f and δ_r , respectively.

4. Simulation

4.1. Simulation Setup

The working process of the vehicle control system is illustrated in Figure 6. The dashed box at the top shows the operation of the vehicle dynamic model. Based on the actual inputs (i.e., T_i and δ_i) as well as the vehicle velocity states vel , the tire model can generate the wheel forces, which is then used in driving unit model. The driving unit model considers the wheel inertia and gives the actual forces acting on the vehicle body. The dynamic states of the vehicle such as positions and velocities are simulated by running the proposed vehicle body dynamic model in Matlab (R2017a, MathWorks, Natick, MA, United States) using a Runge–Kutta method based solver with a time step of 8.33 μ s.

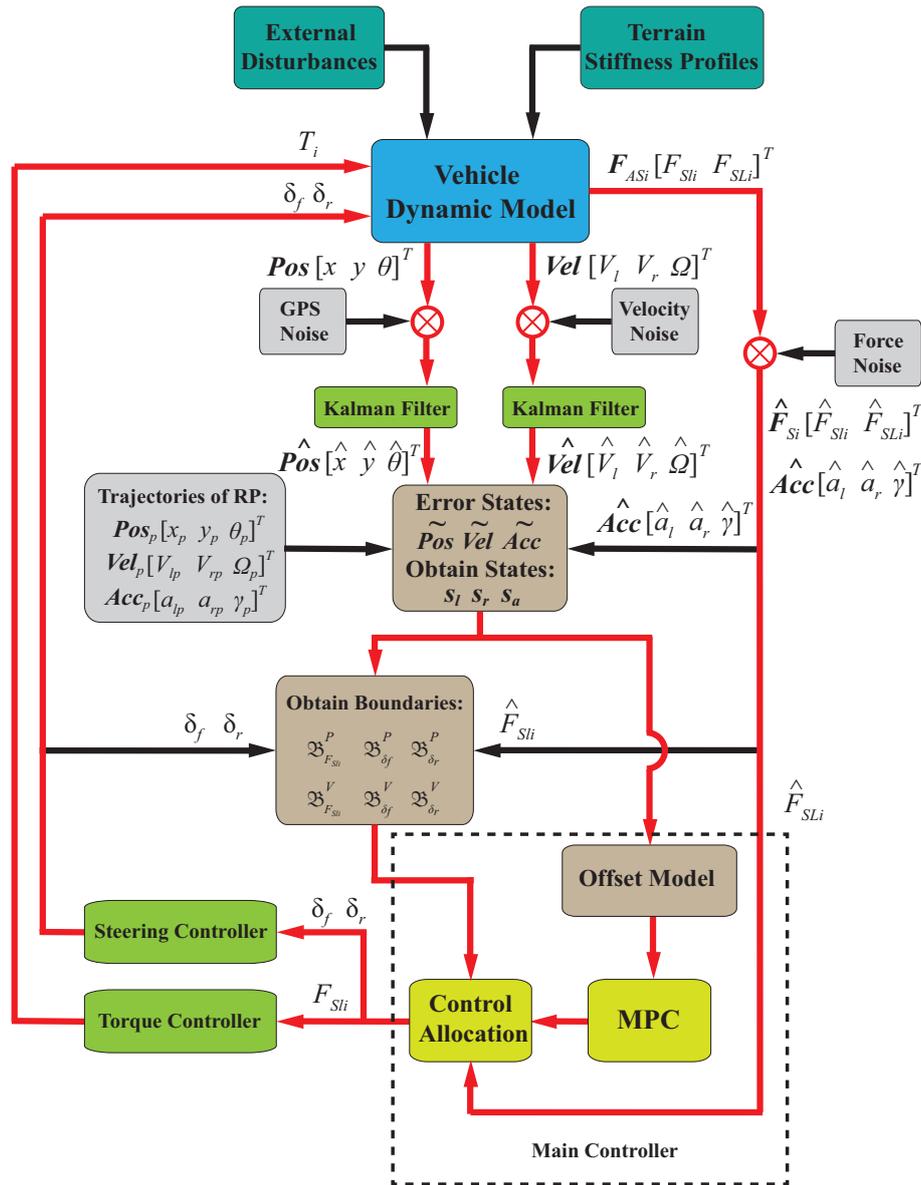


Figure 6. Flowchart of the vehicle control system.

In the simulation, both external disturbances and state measurement noises are involved in validating the robustness of the proposed control methodology. The measured states are compared with the reference profiles to generate the error states. Using the method proposed in Section 3.3, the search boundaries of the drive forces and steering angles are obtained. Meanwhile, the error states are transferred to the offset model. Virtual inputs calculated by the MPC algorithm are delivered to the control allocation module. Then, the actual control inputs including drive forces and steering angles are generated and substituted into the dynamic model for the next iteration. The dashed box at the bottom represents the main controller. PSO controller can be substituted in place of the MPC controller.

The PSO and MPC controllers are applied to drive the simulated vehicle to track the reference path shown in Figure 7a. In the simulation, the varying terrain conditions are considered to verify the validity and robustness of each controller. As presented in Figure 7b, the reference path is divided into ten sections, which have different tire stiffnesses k_{li} and k_{Li} acting on each wheel. To simulate the terrain disturbances, the disturbances F_{gi} applied to all four wheels were always in the vehicle longitudinal direction. As shown in Figure 8, the disturbances are modeled as step signals and their magnitudes are generated randomly to show the uncertainties.

As a comparison, a kinematic model based control method is applied to drive the same vehicle. According to relative kinematic path tracking research [8,19], the side slip is the main disturbance that leads to the unpredicted tracking errors. Using the dynamic model based observer, the side slip of the vehicle can be predicted with an error of 10% to 30%. In this simulation, a random side slip velocity that is less than 10% of its longitudinal velocity is added in the kinematic model.

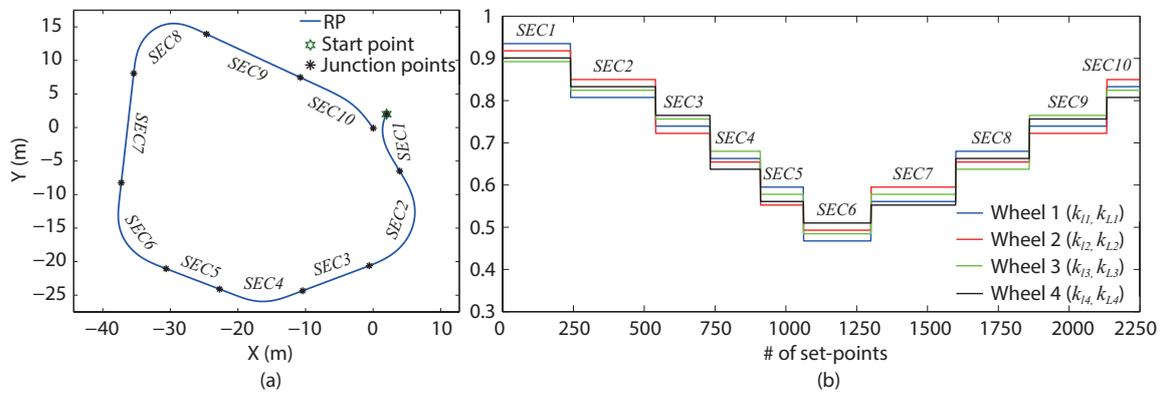


Figure 7. Reference path (RP) and terrain coefficients along the RP. (a) Reference path; (b) terrain coefficients.

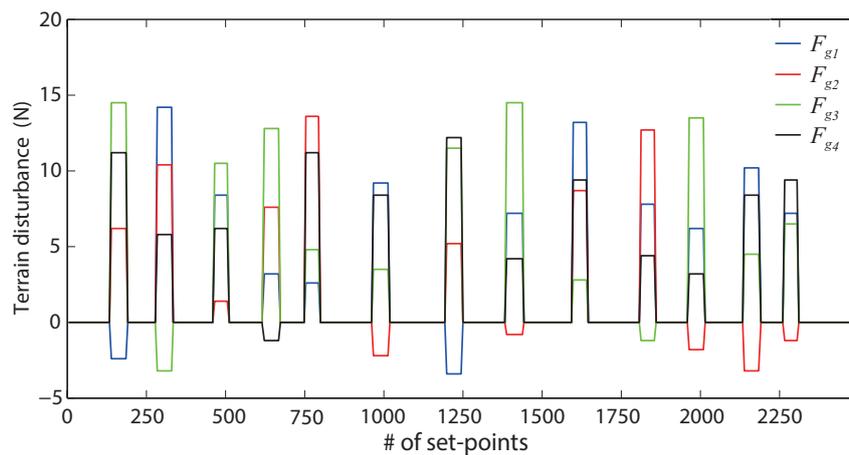


Figure 8. Terrain disturbance.

4.2. Parameters of Simulation

The first two sections of Table 1 list the parameters of vehicles used in simulation. The common parameters in the objective functions are listed in the third section of Table 1, in which the vehicle mass and inertia values used in the objective function are different from actual vehicle values to simulate parametric uncertainties. This helps to verify the robustness of the vehicle control method in dealing with variations in the system. In order to run in real time, both optimization methods are limited to 15 ms of search time per iteration, less than the system sample time $T_s = 20$ ms. In optimization, the optimal solution is searched within the boundaries until time reaches TI or the error gradient of 0.001 is achieved by the objective function. The last two sections present the particular parameters for each method.

Table 1. Parameters used in the simulation.

| Para | Value | Unit | Para | Value | Unit |
|---|-----------------------------|-------------------|-----------------------|---------------------------------|-------------------|
| Vehicle model | | | | | |
| M | 200.0 | kg | m_d | 15.0 | kg |
| L_f | 0.85 | m | L_r | 0.85 | m |
| L_h | 0.5 | m | R_w | 0.25 | m |
| J_z | 45.0 | kg·m ² | J_w | 0.8 | kg·m ² |
| System constraints | | | | | |
| $F_{d\max}$ | 250.0 | N | δ_{\max} | 40.0 | ° |
| $F_{d\min}$ | -250.0 | N | δ_{\min} | -40.0 | ° |
| $\Delta F_{d\max}$ | 0.8 | N | $\Delta\delta_{\max}$ | 0.35 | ° |
| Common parameters in objective function | | | | | |
| M_c | 205.0 | kg | J_{zc} | 40.0 | kg·m ² |
| λ_l | 2.0 | - | λ_r | 1.5 | - |
| λ_a | 2.8 | - | TI | 15 | ms |
| MPC and SQP parameters | | | | | |
| \bar{R} | $0.1 \times I^{3 \times 3}$ | - | Q_f | $1e^{-4} \times I^{4 \times 4}$ | - |
| Q_δ | $0.1 \times I^{2 \times 2}$ | - | Q_s | $I^{3 \times 3}$ | - |
| PSO parameters | | | | | |
| SS | 24 | - | ζ | 0.9 | - |
| ϕ_1 | 1.85 | - | ϕ_2 | 1.85 | - |
| σ_F | 0.4 | - | σ_{δ_f} | 0.5 | - |
| σ_{δ_r} | 0.5 | - | C_l | 0.35 | - |
| C_r | 0.35 | - | C_a | 0.3 | - |

$I^{i \times i}$ represents the i -th-order identity matrix. MPC: model predictive control; SQP: sequential quadratic programming; PSO: particle swarm optimization.

4.3. Simulation Results

Figure 9 shows path tracking results by different controllers. The MPC-SQP controller has superior performance with offset error less than 3.2 cm, and heading error less than 2°. The PSO controller has maximum offset error of 6.1 cm and heading error of 3.2°. Both dynamic model based control methods perform better than the kinematic model based one in constraining the offset errors.

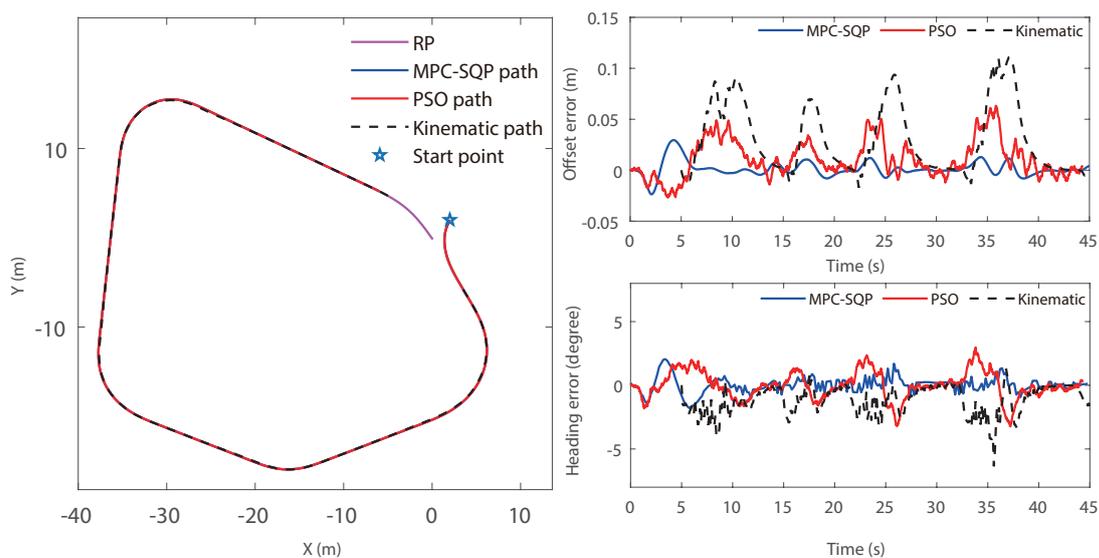


Figure 9. Path following performance.

The drive torques and steering angles applied on four wheels are presented in Figure 10. At each cornering, the torques applied at the two outside wheels (i.e., T_1 and T_4) firstly increase and then decrease, which are contrary to that of the two inside wheels (i.e., T_2 and T_3). The difference in torques is used to compensate the insufficient angular accelerations of vehicle in corners. The steering angles of MPC-SQP and PSO controller follow the same varying trend in which steering curves of MPC-SQP show smoother variation. Thus, the MPC-SQP controller can provide more stable steerings compared with the PSO controller.

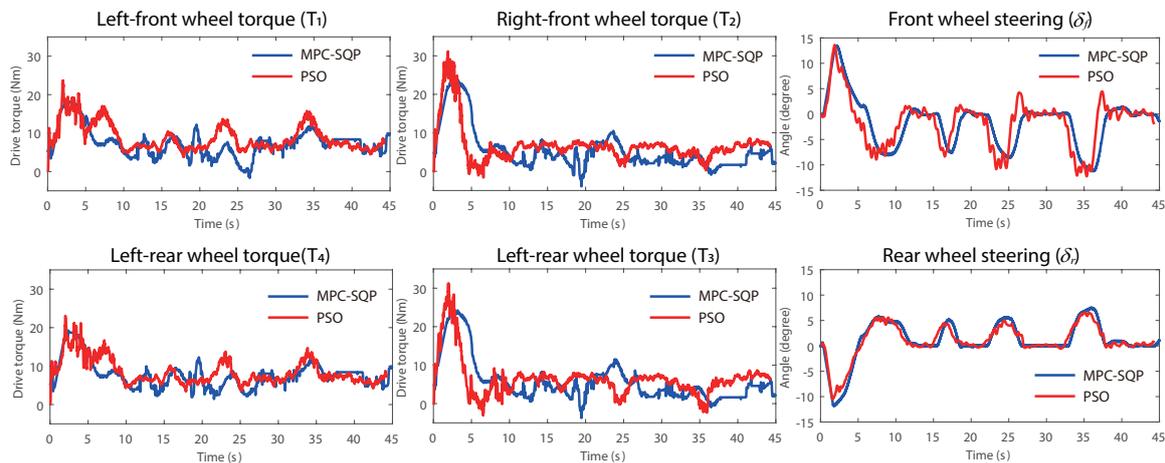


Figure 10. Actual inputs.

In this work, another aim of control is to maintain high vehicle motion quality including its velocity and acceleration performances. Figure 11a presents the longitudinal velocity curves achieved by MPC-SQP and PSO controllers. Starting at 0.5 m/s, the two curves reach 3 m/s at 6.5 s and 4.8 s, respectively. Thus, the trajectory and error curves of the MPC controller responses are slower than the PSO controller in Figure 9. Both controllers are capable of maintaining the longitudinal velocity around 3 m/s in the rest process. In Figure 11b, the longitudinal acceleration curve of PSO increases faster than MPC-SQP and has a peak value of 1 m/s². The curve of MPC-SQP controller reaches 0.6 m/s² at a maximum and has smoother variation during the whole process.

From the offset model, lateral velocity is undesirable as it causes tracking error. As shown in Figure 11c, both controllers maintain the lateral velocities fluctuating around zero. The MPC-SQP controller has a maximum lateral velocity of 0.08 m/s, whereas PSO has a maximum value of 0.17 m/s. The lateral velocities are well constrained, which provides higher accuracy of path tracking. The lateral acceleration curves follow the same varying trend in Figure 11d. It can be seen the PSO controller causes a lateral acceleration oscillation of 0.6 m/s² relative to the MPC-SQP. The smoother change of lateral acceleration achieved by MPC-SQP controller reduces the jerk effect, decreasing the deviation from the reference path. The angular velocities are demonstrated in Figure 11e, in which the MPC-SQP has higher accuracy and a smoother manner, compared with that of the PSO. In Figure 11f, the angular acceleration curves are presented. MPC-SQP and PSO curves vary with oscillations of 12 °/s² and 20 °/s², respectively. The undesirable oscillations are eliminated in the process of the integration, which can be demonstrated in Figure 11e. From all the acceleration plots given in Figure 11, the MPC-SQP controller performs better in constraining oscillations of accelerations, which provides better stability in vehicle motion control.

Considering that the optimization based method may lead to an expensive computation, it is essential to analyze the computing efficiency for each controller. As shown in Figure 12, the computing efficiency of each controller is compared. Figure 12a shows the computing time used by the controllers in each sampling time T_s . It can be seen that both controllers achieve finishing computing within T_s , which validates the feasibility of proposed controllers. In the box plot in

Figure 12b, both controllers have an average computing time of 8 ms. The PSO controller gives a stable variation range from 4 ms to 10 ms, as the particle velocity vector is decided by the limitation of the sampling time, which guarantees relatively high quality solutions within a short time. The computing time of MPC-SQP controller substantially increases and reaches 15 ms to 20 ms during each cornering because the control allocation may encounter a complex optimization problem when the lateral forces start to vary. It indicates that the PSO controller performs better in computing efficiency.

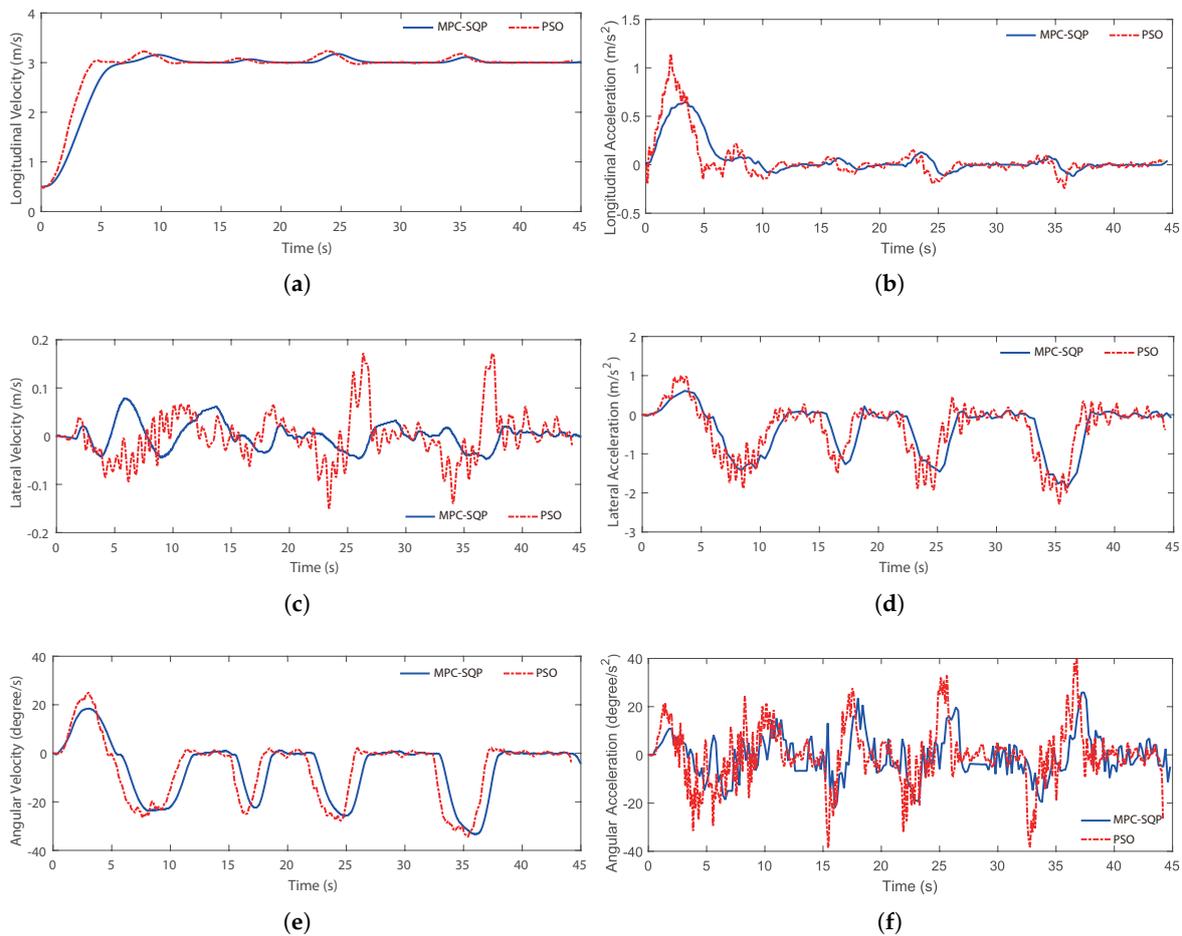


Figure 11. Acceleration and velocity performances. (a) longitudinal velocity; (b) longitudinal acceleration; (c) lateral velocity; (d) lateral acceleration; (e) angular velocity; (f) angular acceleration.

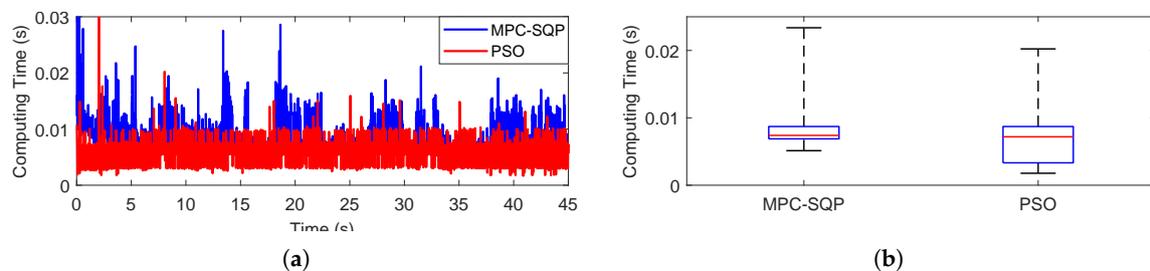


Figure 12. Computing efficiency comparison. (a) computing time; (b) box plot of computing time.

5. Discussion

Based on the theoretical analysis and simulation results provided above, the discussion between MPC based method and PSO method are given as follows:

- D1 Comparing the simulations, it is obvious that dynamic-based methods proposed in this paper perform better than kinematic model based methods. In the kinematic model based methods, it is difficult to measure the side slip directly. The majority of research works are trying to design observers to predict its side slip. However, this kind of method can only get an approximate estimate. Thus, it is not feasible to use a kinematic model based controller to completely eliminate the error due to side slip. Both MPC-SQP and PSO methods are proposed based on dynamic models and controlled by drive forces. In the dynamic model, the lateral forces can be obtained easily using force sensors. This kind of method gives a practical way to avoid the side slip estimation and achieve precise tracking along curved paths.
- D2 Both MPC-SQP and PSO methods are proposed partly or completely based on optimizations. As is compared in the simulation, the PSO controller achieves obtaining the solution with a stable computing time. The algorithm can optimize the quality of solutions and the calculation times at the same time, which guarantees the feasibility and capability of the controller. The MPC-SQP controller may encounter the calculation timeout in the simulation results. This is because the SQP solver spends more time when calculating a complex Hessian matrix. Thus, it is necessary to set a sufficient sampling time when applying the MPC-SQP method.
- D3 The MPC-SQP method gives a stability proof of the control system while the PSO method has difficulty with the mathematical proof due to the limitation of intelligent optimization. The stability analysis of MPC-SQP method provides a possibility to analyze the stable range of its control parameters and margin of errors.
- D4 Both control algorithms are reduced to constrained optimization problems with six input variables. PSO is a global algorithm that performs better with searching for a global optimal solution while SQP is a reliable solver but may be held in local optimal solutions. Thus, for each method, it is important to define a proper search range. In this paper, the boundaries are calculated based on the vector s , which improves the efficiency and stability of the optimization process.

6. Conclusions

This paper presented two control methodologies applicable to four wheel steering and four wheel drive vehicle systems to track paths accurately. The system model is a nonlinear coupled dynamic model that is over-actuated. The first methodology determines the drive force inputs and steer inputs using an MPC based method coupled with a control allocation based on SQP. The second methodology is proposed as a fully optimization based method that is developed by refining a previously proposed PSO based method. The performance of the MPC-SQP method has been compared with the PSO based control method and a kinematic model based control method. In the simulation, the path tracking results have proved the superior performance of the MPC-SQP based controller. In the cases when computing times are considered, the PSO based controller offers more efficiency and better stability in computing compared with MPC-SQP based controller.

Author Contributions: Conceptualization, Q.T. and J.K.; Methodology, Q.T. and P.D.; Writing—Original Draft Preparation, Q.T. and Z.Z.; Writing—Review & Editing, J.K.; Supervision, J.K.; Project Administration, J.K.; Funding Acquisition, Q.T.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, Grant No. 2017JBM051 and 2015JBC022, and the China Postdoctoral Science Foundation, Grant No. 2016M600910.

Acknowledgments: This work was carried out at the School of Mechanical and Manufacturing Engineering, the University of New South Wales, Sydney, Australia.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the writing of the manuscript.

References

1. Oppenheimer, M.W.; Doman, D.B.; Bolender, M.A. Control Allocation for Over-actuated Systems. In Proceedings of the 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 28–30 June 2006; pp. 1–6.
2. Ackermann, J.; Sienel, W. Robust yaw damping of cars with front and rear wheel steering. *IEEE Trans. Control Syst. Technol.* **1993**, *1*, 15–20. [[CrossRef](#)]
3. Itoh, H.; Oida, A.; Yamazaki, M. Numerical simulation of a 4WD-4WS tractor turning in a rice field. *J. Terramech.* **1999**, *36*, 91–115. [[CrossRef](#)]
4. Wu, J.; Wang, Q.; Wei, X.; Tang, H. Studies on improving vehicle handling and lane keeping performance of closed-loop driver-vehicle system with integrated chassis control. *Math. Comput. Simul.* **2010**, *80*, 2297–2308. [[CrossRef](#)]
5. Wang, W.; Song, Y. A new high dimension nonlinear dynamics simulation model for four-wheel-steering vehicle. *J. Mech. Eng. Sci.* **2013**, *227*, 29–37. [[CrossRef](#)]
6. Udengaard, M.; Iagnemma, K. Kinematic analysis and control of an omnidirectional mobile robot in rough terrain. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, San Diego, CA, USA, 29 October–2 November 2007; pp. 795–800.
7. Grand, C.; Amar, F.B.; Plumet, F. Motion Kinematic analysis of wheeled-legged rover over 3D surface with posture adaptation. *Mech. Mach. Theory* **2010**, *45*, 477–495. [[CrossRef](#)]
8. Han, G.; Fu, W.; Wang, W.; Wu, Z. The Lateral Tracking Control for the Intelligent Vehicle Based on Adaptive PID Neural Network. *Sensors* **2017**, *17*, 1244. [[CrossRef](#)] [[PubMed](#)]
9. Qian, H.; Lam, T.; Li, W.; Xia, C.; Xu, Y. System and design of an Omni-directional vehicle. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Bangkok, Thailand, 22–25 February 2009; pp. 389–394.
10. Grepl, R.; Vejlupek, J.; Lambersky, V.; Jasansky, M.; Vadlejš, F.; Coupek, P. Development of 4WS/4WD Experimental Vehicle: Platform for research and education in mechatronics. In Proceedings of the IEEE International Conference on Mechatronics (ICM), Istanbul, Turkey, 13–15 April 2011; pp. 893–898.
11. Ramaswamy, S.A.P.; Balakrishnan, S.N. Formation control of car-like mobile robots: A Lyapunov function based approach. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 657–662.
12. Camacho, E.; Alba, C. *Model Predictive Control*; Advanced Textbooks in Control and Signal Processing; Springer: London, UK, 2013.
13. Lenain, R.; Thuilot, B.; Cariou, C.; Martinet, P. Model predictive control for vehicle guidance in presence of sliding: Application to farm vehicles path tracking. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), Barcelona, Spain, 18–22 April 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 885–890.
14. Lenain, R.; Thuilot, B.; Cariou, C.; Martinet, P. High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks. *Auton. Robots* **2006**, *21*, 79–97. [[CrossRef](#)]
15. Prasetya, D.A.; Yasuno, T. Cooperative control of multiple mobile robot using particle swarm optimization for tracking two passive target. In Proceedings of the SICE Annual Conference (SICE), Akita, Japan, 20–23 August 2012; pp. 1751–1754.
16. Lin, L.; Sun, Q.; Wang, S.; Yang, F. Research on PSO based multiple UAVs real-time task assignment. In Proceedings of the Chinese Control and Decision Conference (CCDC), Guiyang, China, 25–27 May 2013; pp. 1530–1536.
17. Thomas, J. Integrating Particle Swarm Optimization with Analytical Nonlinear Model Predictive Control for nonlinear hybrid systems. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, France, 21–23 July 2015; pp. 294–301.
18. Dai, P.; Katupitiya, J. Force control for path following of a 4WS4WD vehicle by the integration of PSO and SMC. *Veh. Syst. Dyn.* **2018**, 1–35. [[CrossRef](#)]
19. Wang, X.; Taghia, J.; Katupitiya, J. Robust Model Predictive Control for Path Tracking of a Tracked Vehicle with a Steerable Trailer in the Presence of Slip. *IFAC-PapersOnLine* **2016**, *49*, 469–474. [[CrossRef](#)]
20. Rajamani, R. *Vehicle Dynamics and Control*; Springer Science: Berlin, Germany, 2006.

21. Smith, D.; Starkey, J. Effects of Model Complexity on the Performance of Automated Vehicle Steering Controllers: Model Development, Validation and Comparison. *Veh. Syst. Dyn.* **1995**, *24*, 163–181. [[CrossRef](#)]
22. Brach, R.; Brach, R. *Tire Models for Vehicle Dynamic Simulation and Accident Reconstruction*; SAE Technical Paper; Society of Automotive Engineers: Warrendale, PA, USA, 2009; Volume 1.
23. Jazar, R.N. *Vehicle Dynamics: Theory and Applications*; Springer: New York, NY, USA; London, UK, 2008.
24. Milliken, W.F.; Milliken, D.L. *Race Car Vehicle Dynamics*; Society of Automotive Engineers: Warrendale, PA, USA, 1995.
25. Dai, P.; Katupitiya, J. Force Control of a 4WS4WD Vehicle for Path Tracking. In Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, Korea, 7–11 July 2015; pp. 238–243.
26. Johansen, T.A.; Fossen, T.I. Control allocation—A survey. *Automatica* **2013**, *49*, 1087–1103. [[CrossRef](#)]
27. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Upper Saddle River, NJ, USA, 1991.
28. Eberhart, R.C.; Shi, Y. Particle swarm optimization: Developments, applications and resources. *Evol. Comput.* **2001**, *1*, 81–86.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).