Research Article

Muhammad Ahsan* and Mostafa M. Salah

Improved nonlinear model predictive control with inequality constraints using particle filtering for nonlinear and highly coupled dynamical systems

Keywords: nonlinear model predictive control, motion planning based on samples, Bayesian estimation, constrained optimization problem, linearization, Monte Carlo sample

1 Introduction

Model predictive control (MPC) has evolved as an extremely effective method for dynamic control and optimization. MPC uses a dynamic model and online measurements to estimate future output behavior as a function of present and future controlled input actions [1,2]. Based on the prediction, optimization is carried out to determine a set of input actions that minimize a selected measure of the deviation of the output from their respective reference values while meeting all specified constraints. This set of input actions minimizes the deviation between selected output items from their respective reference values. Since prediction accuracy can improve with additional samples, only the first calculated input sequence is used and the entire optimization is performed at the next sampling interval. This “receiving-horizon implementation” turns MPC into a feedback-control approach [3,4]. A crucial aspect that contributes to MPC’s effectiveness is the ability to include diverse process constraints directly into the online optimization performed at every time interval [5,6].

The optimization problem, which consists of input controller and state variables, is solved over a finite time horizon within the MPC approach. The state trajectories are predicted by online merging the local plans, and then, these state trajectories are used for control applications in motion planning [7]. The point-to-point control technique is used to solve the control problems in linear MPC scenarios. Ghazaei Ardakani et al. [8] proposed a point-to-point linear MPC approach for trajectory tracking problems. In the study by Neunert et al. [9], a nonlinear
control problem is solved using an iterative sequential linear quadratic technique and optimized control inputs for feedforward and feedback control are generated with the application of a hex copter vehicle.

MPC was improved to solve the problem of nonlinear control for industrial processes in the late 1970s, and the resulting modified controller was named nonlinear model predictive control (NMPC) [1,2,10]. NMPC is a technology that has gained popularity in business and academics for constructing and monitoring collision-free trajectories in real time for autonomous vehicle systems and vehicle robotics [11–13]. NMPC is a feedback control approach that is used to develop optimal control actions for highly nonlinear systems such as autonomous cars. It entails forecasting the future behavior of the system using a mathematical model while accounting for system dynamics as well as internal and external limitations [14]. This enables the controller to provide a series of control inputs that will move the system to the target state while avoiding accidents and other risks. One of the benefits of NMPC is its ability to handle nonlinear systems with restrictions, which is vital for autonomous cars operating in dynamic and unpredictable situations. Another benefit of NMPC is that it is a predictive control approach, which means that it can anticipate changes in the environment and adapt the vehicle’s trajectory in real time. This makes NMPC ideal for self-driving cars that must continually adjust to changing situations [11–13–16].

NMPC is known as the moving horizon or predictor horizon using a system dynamics model. The moving horizon or predictor horizon refers to the fact that the controller updates its predictions at every time step as new data becomes available. The controller computes a new optimal input sequence at each time step, which takes into account the current state of the system as well as any new data that have become available since the previous time step [14,17,18]. The use of a dynamical model in NMPC allows the controller to take into account the nonlinear dynamics of the system and any constraints on the inputs or outputs of the system. By generating a sequence of control inputs over a prediction horizon, the controller is able to anticipate changes in the system and adapt its control actions accordingly [13,14,19]. Feedback control is optimized to solve the control problem for a nonlinear system in NMPC [20]. In the NMPC method, the state and input restrictions can be changed while still respecting the system dynamics nonlinearities and coupling.

NMPC has many applications in control engineering, including setpoint stabilization and route following or trajectory tracking. Setpoint stability can be used to solve the fluid-level control problem or temperature control problem [21,22]. Trajectory tracking is a complex task in which reference trajectory changes with time. Hovorka et al. [23] and Aguiar and Hespanha [24] provide a number of examples from the disciplines of autonomous drones and healthcare. Meanwhile, time-invariant reference trajectories are the focus of the route-following problem, with the objective of achieving optimal route tracking independent of time. In Matschek et al. [25] and Faulwasser et al. [26], NMPC is applied to solve the trajectory-tracking problem in which a robot manipulator is proposed to track the predetermined reference trajectory. Matschek et al. [27] gave an in-depth description of the problems associated with setpoint stabilization, trajectory tracking, and route following, as well as their related characteristics and obstacles.

This study introduces a significant contribution by seamlessly integrating a sampling-based motion planning algorithm with NMPC through Bayesian filtering. This novel approach enables the real time solution of nonlinearly constrained optimization problems, specifically tailored for the task of vehicle robot path tracking. Additionally, the study introduces particle filtering, a powerful algorithm designed to implement Bayesian filtering in nonlinear and non-Gaussian systems. This technique, operating as a sequential Monte Carlo (SMC) method, effectively represents the posterior distribution by using a set of weighted particles. These particles are iteratively propagated through time via a recursive algorithm, providing an accurate estimation of system states. Moreover, this study proposes a method for estimating optimal control actions using Bayesian filtration, drawing from reference signals with predefined upper and lower boundaries. This approach empowers precise control decisions, even in scenarios fraught with uncertainties and constraints. Furthermore, the research emphasizes the efficiency of the SMC sampling method, significantly enhancing control actions by conducting a thorough search in an expansive control space facilitated by a large number of particles. This is particularly advantageous in scenarios characterized by complex or high-dimensional solution spaces. The study concludes by demonstrating the effectiveness and simplicity of the proposed sampling-based NMPC technique, particularly in the context of nonlinear and strongly coupled dynamical systems. This approach represents an effective tool for practical applications demanding precise control and estimation. Overall, these contributions collectively advance the state of the art in the seamless integration of sampling-based motion planning, NMPC, and Bayesian filtering, particularly in scenarios involving nonlinear and non-Gaussian systems. This work not only furthers theoretical foundations but also offers practical solutions for real time applications in vehicle robotics and related fields.
Following is an expanded explanation of how our work contributes to the state of the art, particularly in the implementation of the particle filtering algorithm in Bayesian filtering for nonlinear and non-Gaussian systems:

- Our study introduces a novel approach that integrates a sampling-based motion planning algorithm with NMPC through Bayesian filtering. This is a significant advancement as it enables real time, online solution of the nonlinearly constrained optimization problem. This integration is particularly tailored for the application of path tracking in vehicle robotics.
- We emphasize the utilization of particle filtering, a powerful algorithm for implementing Bayesian filtering in systems characterized by nonlinearity and non-Gaussian distributions. This approach uses an SMC method to represent the posterior distribution. This is a crucial contribution, as it allows for accurate estimation in scenarios where traditional linear methods may fall short.
- Our work highlights the core principle of particle filtering, which involves representing the posterior distribution using a set of weighted particles. These particles are recursively propagated through time using an algorithm. This technique ensures that the estimation process is continually refined, providing accurate and up-to-date information about the system’s state.
- One of the key innovations of our work lies in the estimation of optimal control actions. This is achieved through Bayesian filtering using reference signals with upper and lower boundaries. By leveraging this approach, we are able to make informed decisions regarding control actions, even in the presence of uncertainties and constraints.
- We highlight the efficiency of the SMC sampling method. This technique significantly enhances the control action by exploring a vast control space using a large number of particles. This is particularly valuable in scenarios where the solution space is complex or high-dimensional.
- Our work demonstrates the effectiveness and simplicity of the proposed sampling-based NMPC technique, especially in the context of nonlinear and strongly coupled dynamical systems. This approach represents a powerful tool for real-world applications where accurate control and estimation are crucial.

The rest of this article is arranged as follows: Section 2 includes the proposed method of NMPC using particle filtering for nonlinear and highly coupled dynamical systems. In this section, first, the NMPC problem is formulated as an optimization problem for a nonlinear system with constraints. Second, the Bayesian estimation approach is established that consists of forward filtering and backward smoothing framework for NMPC. Third, the particle filtering approach is proposed to estimate the posterior probability distribution for the closed-loop nonlinear system. Fourth, the penalty function is introduced in the particle filtering approach to enhance the NMPC approach so that it can tackle the inequality constraints. Section 3 illustrates the simulation results that construct that dynamical model of a discrete-time nonlinear model for a robot vehicle and the tracking problem is simulated using the proposed NMPC approach. The simulation results show that the proposed NMPC approach using particle filtering incorporating barrier function solves the tracking problem efficiently.

2 Proposed method

2.1 NMPC problem formation

In order to develop the NMPC problem, first we introduce the discrete-time nonlinear system and then construct the cost function using the error signal. The discrete-time nonlinear model at time \( \tau \) subjected to the \( n \) number of inequality constraints can be written as follows [19,20]:

\[
\xi_{i+1} = f(\xi_{i}, \psi_{i}),
\]

\[
g_i(\xi, \psi) \leq 0, \quad \forall i = 1, ..., n,
\]  

where \( \xi \) and \( \psi \) are the state of the system and the control signal at time \( \tau \), and \( f(\cdot) \) and \( g(\cdot) \) are the nonlinear mapping functions, and \( n \) is the total number of constraints for the nonlinear discrete-time system.

The object of the problem is to design a control signal such that the error signal is minimized, and hence, the state of the nonlinear discrete-time system tracks the ideal reference signal. If the reference signal is represented by \( \tilde{\xi}_{\tau} \), then the error signal can be written as \( e_{i} = \xi_{i} - \tilde{\xi}_{\tau} \). The cost function is given as follows:

\[
J(e_{i}, \psi_{i}, T) = \sum_{i=1}^{T} \|e_{i}\|^{2}_{W_{1}} + \|\psi_{i}\|^{2}_{W_{2}},
\]  

where \( T \) is the length of the horizon upcoming, \( \psi_{\tau:T+T} = [\psi_{\tau}, \psi_{\tau+1}, ..., \psi_{\tau+T}] \), and \( W_{1} \) and \( W_{2} \) are weighting matrices. Furthermore, the cost function is the weighted quadratic sum of the tracking error and the control cost. The cost function given in Eq. (2) is subjected to the nonlinear discrete-time model of Eq. (1) and forms the NMPC problem as follows:

\[
\min_{\psi_{\tau:T}} J(e_{i}, \psi_{i}, T)
\]

s.t. \( \xi_{i+1} = f(\xi_{i}, \psi_{i}) \),

\[
g_i(\xi_{i}, \psi_{i}) \leq 0, \quad \forall i = 1, ..., n,
\]

\( t = \tau, ..., \tau + T \).
The NMPC problem represented in Eq. (3) forms an optimization problem that is solved to design the optimal control signals \( \psi^{*}_{t:t+T} \) and optimize the cost function for the control signals over the \( T \) horizon. When the optimization problem is solved and the optimized control signals are evaluated, the first element of the optimized control signals \( \psi^{*}_{t} \) is applied and all other control elements are discarded. The process is repeated for all steps in \( T \) and forms a model-based predictive optimization problem [17,19].

Alternatively, the NMPC problem can also be represented as an optimization problem in which states of the nonlinear discrete-time model and its control signals are estimated using the desired reference states. The overall model can be represented as follows:

\[
\begin{align*}
\xi_{t+1} &= f(\xi_{t}, \psi_{t}), \\
\psi_{t+1} &= \omega_{t}^{(1)}, \\
\xi'_{t} &= \xi_{t} + \omega_{t}^{(2)},
\end{align*}
\]

(4)

where \( \omega_{t}^{(1)} \) and \( \omega_{t}^{(2)} \) are the additive disturbances that form the cost function as follows:

\[
f(\omega_{t}^{(1)}, \omega_{t}^{(2)}, T) = \sum_{t=\tau}^{t+T} ||\omega_{t}^{(1)}||^{2}_{\text{hl}} + ||\omega_{t}^{(2)}||^{2}_{\text{hl}}.
\]

(5)

The state and control signal for the NMPC problem represented in Eq. (4) can be estimated using the following optimization problem:

\[
\begin{align*}
\min_{\xi_{t:t+T}, \psi_{t:t+T}} & f(\omega_{t}^{(1)}, \omega_{t}^{(2)}, T) \\
\text{s.t.} & \xi_{t+1} = f(\xi_{t}, \psi_{t}), \\
& \psi_{t+1} = \omega_{t}^{(1)}, \\
& \xi'_{t} = \xi_{t} + \omega_{t}^{(2)}, \\
& g(\xi_{t}, \psi_{t}) \leq 0, \quad \forall i = 1, ..., n, \\
& t = \tau, ..., T + \tau.
\end{align*}
\]

(6)

The optimization problem described in Eq. (6) is similar to the optimization problem formed in Eq. (3), which established a new way to deal with the NMPC problem.

### 2.2 Bayesian estimation approach for NMPC

Bayesian estimation is a statistical method for estimating the parameters of the model based on previous information and observed data. It is commonly used in control systems to increase the accuracy and resilience of the control algorithm, particularly in nonlinear systems [28,29]. A typical way for applying Bayesian estimation is the forward filtering and backward smoothing approach. A Bayesian

forward filtering and backward smoothing framework for NMPC involves estimating the posterior probability distribution over the state space at each time step based on observed data and prior knowledge and using this estimate to predict the future behavior of the system and optimize the control actions.

In the NMPC problem, the future trajectories of the states with optimized control inputs \( (\xi_{k:t}, \psi_{k:t}) \) are estimated from the reference trajectory \( \xi'_{k:t} \). The forward filtering stage entails predicting the system's state based on observable data up to the present moment. This is accomplished by estimating the posterior probability distribution of the state based on observations and previous knowledge of system dynamics \( \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t}) \) given the observations up to time \( k \leq t \leq k + H \) as follows:

\[
\rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t}) = \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t-1}, \psi_{k:t-1}) \rho(\xi'_{k:t-1}, \psi_{k:t-1} | \xi'_{k:t-1}),
\]

(7)

Eq. (7) shows the relationship for the probability distribution of forward trajectories \( \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t}) \) from previous trajectories \( \rho(\xi_{k:t-1}, \psi_{k:t-1} | \xi'_{k:t-1}) \). The state estimation is then used to decide the control action. The backward smoothing stage from \( \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t+1}) \) to \( \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t-1}) \) is then used to refine the state estimate by adding all relevant data, both past and future. This is performed by calculating the posterior probability distribution of the state given all of the observations and prior knowledge of the system dynamics:

\[
\rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t+1}) = \int \rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t-1}) \frac{\rho(\xi_{k:t-1}, \psi_{k:t-1} | \xi'_{k:t-1})}{\rho(\xi_{k:t}, \psi_{k:t} | \xi'_{k:t-1})} d(\xi_{k:t-1}, \psi_{k:t-1}).
\]

(8)

Bayesian estimate necessitates computing the posterior probability distribution across the state space, which is typically not available in closed-up nonlinear systems. One way to overcome this difficulty is to use nonlinear versions of the Kalman filter to estimate the posterior distribution of the state given the observations and the model [30,31]. Their accuracy, however, is dependent on the model's quality and the system's complexity. Another way is to use particle filters, which are a type of Monte Carlo method that uses a collection of weighted samples to approximate the posterior distribution. Particle filters can handle nonlinear models and non-Gaussian distributions.

### 2.3 Particle filtering approach for NMPC

The idea of importance weights is fundamental to SMC sampling, commonly known as particle forward filtering and backward smoothing [28,32]. SMC approaches are used
to iteratively estimate the posterior distribution of a state space model based on a series of observations. The importance weights are used to scale particles from the importance distribution in the particle forward filtering and backward smoothing. The importance distribution is chosen to be the empirical distribution of the particles created in the previous time step. Importance weights are used to weight the particles, and they are calculated as the ratio of the target distribution (the posterior distribution) to the importance distribution assessed at the particle values [28,32].

Formally, let $\xi_{k:t}$ be the state variable, $\xi^j_{k:t}$ be the observation variable, and $\psi_{k:t}$ be a set of parameters to be estimated. The target distribution is the joint posterior distribution (unknown distribution) of $\xi_{k:t}$ and $\psi_{k:t}$ given the observations up to time $k \leq t \leq k + H$: $\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})$. The importance distribution (known distribution) is the proposal distribution used to generate the particles: $q(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})$. The importance weights drawn from $q$, are given by:

$$w_i^{(1)} = \frac{\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})}{q(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})}. \quad (9)$$

The weights are then normalized so that they sum to one, and the particles are resampled based on their weights to generate a new set of particles for the next time step. Eq. (9) can be written as follows using the approximation $\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}) = \sum_i w_i^{(1)} \delta(\xi_{k:t}, \psi_{k:t} - \xi^k_{i:t}, \psi^{(i)}_{k:t})$:

$$w_i^{(f)} = \frac{\rho(\xi^j_{k:t}, \psi^{(i)}_{k:t}) \rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k:t})}{q(\xi^j_{k:t}, \psi^{(i)}_{k:t})} \cdot \frac{\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k:t})}{\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})}. \quad (10)$$

Eq. (10) shows that the importance weights $w_i^{(f)}$ are updated recursively. Furthermore, consider $q(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k:t}) = \rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k:t})$, and samples $(\xi^{(i)}_{k:t}, \psi^{(i)}_{k:t}) = \rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})$, are taken at time $t$. Then, Eq. (10) is updated as follows:

$$w_i^{(f)} = \frac{\rho(\xi^j_{k:t}, \psi^{(i)}_{k:t})}{\sum_j \rho(\xi^{(j)}_{k:t}, \psi^{(j)}_{k:t})} \cdot \frac{\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k:t})}{\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t})}. \quad (11)$$

The derived equation is called particle forward filtering. Particle degeneracy is a typical issue in particle forward filtering in which the weights of the particles are either zero in the beginning or become zero after some epochs, resulting in poor estimate efficiency. Resampling is an effective strategy for dealing with this issue. Resampling creates a new collection of particles from an existing set of particles depending on their weights [33,34]. Particles with greater weights are more likely to be chosen, whereas those with lower weights are less likely. The number of particles with non-zero weights is raised by resampling, and the overall quality of the particle collection is enhanced.

Backward smoothing gives an improved estimation of samples. In Eq. (8), $\rho(\xi_{k+1:t}, \psi_{k+1:t} | \xi^j_{k:t})$ is replaced with $\int \rho(\xi_{k+1:t}, \psi_{k+1:t} | \xi^j_{k:t}, \psi^{(i)}_{k}) \rho(\xi^j_{k:t}, \psi^{(i)}_{k}) d(\xi^j_{k:t}, \psi^{(i)}_{k})$, and the resultant expression is written as follows:

$$\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k+H:T}) = \rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}) \times \int \rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k:t}, \psi^{(i)}_{k}) \rho(\xi^j_{k:t}, \psi^{(i)}_{k}) d(\xi^j_{k:t}, \psi^{(i)}_{k}) \times \rho(\xi^j_{k+1:t}, \psi^{(i)}_{k+1:t} | \xi^j_{k:t+1:T}) d(\xi^j_{k+1:t}, \psi^{(i)}_{k+1:t}). \quad (12)$$

From Eq. (12), it can be seen that all the posterior distributions can be approximated with importance distribution, and consequently, the samples for backward smoothing can be given as follows:

$$w_{k+1}^{(f)} = \frac{\rho(\xi^j_{k+1:T}, \psi^{(i)}_{k+1:T} | \xi^j_{k+H:T})}{\sum_i w_{k+1}^{(f)}(\xi^j_{k+1:T}, \psi^{(i)}_{k+1:T})}. \quad (13)$$

The posterior distribution in Eq. (12) is approximated as follows:

$$\rho(\xi_{k:t}, \psi_{k:t} | \xi^j_{k+H:T}) \approx \sum_{i=1}^N w_{k+1}^{(f)}(\xi^j_{k+H:T}, \psi^{(i)}_{k+1:T}), \quad (14)$$

which leads to the best estimation of $(\xi^*_k, \psi^*_k)$ from $\xi^j_{k+H:T}$ and optimal control inputs are given at $k$ time step as follows:

$$(\xi^*_k, \psi^*_k) = \sum_{i=1}^N w_{k+1}^{(f)}(\xi^j_i, \psi^{(i)}_k). \quad (15)$$

In the aforementioned method for the NMPC algorithm, the weights are improved to enhance the backward smoothing. However, it is important to note that this method does not take into account inequality constraints. As a result, it is classified as the normal-NMPC approach. Next, we will consider the constraints and develop an improvement in the proposed method for the NMPC algorithm.

### 2.4 Particle filtering approach for NMPC with inequality constraints

Incorporating inequality constraints into particle forward filtering and backward smoothing can be challenging, as traditional particle forward filtering and backward smoothing may not be able to handle constraints directly. A barrier function is a mathematical function that is used in optimization problems to enforce constraints on the variables being optimized [35]. In NMPC, barrier functions are used to ensure that the system being controlled satisfies constraints on its
state and inputs over a finite time horizon. The basic idea behind using a barrier function in NMPC is to penalize the likelihood of particles that violate the constraints, while still allowing the particles to explore the entire state space [36–38]. This is achieved by introducing a penalty term that grows infinitely large as the particles approach the constraint boundary, effectively creating a “barrier” that prevents the particles from crossing the boundary. To implement this approach, we would first define a virtual function as follows:

\[ v_t = f_b(g(\xi_t, \psi_t)) + n_t, \]  

where \( v_t \) is the state of the virtual function, \( f_b \) is a barrier function, \( g(\xi_t, \psi_t) \) are the inequality constraints defined in Eq. (6), and \( n_t \) is the random noise with a zero mean.

The barrier function should return a value that increases as the constraints are violated and approaches infinity as the constraints are approached. We consider the following barrier function [37,38]:

\[ f_b(s) = \frac{1}{k_1} \ln(1 + \exp(k_2 s)), \]  

where \( k_1 \) and \( k_2 \) are constants and \( s \) is a variable on which barrier function depends.

Finally, we would incorporate these virtual measurements into the particle forward filtering and backward smoothing algorithm. This would allow the algorithm to take into account the constraints in the estimation process, while still allowing the particles to explore the entire state space. Eq. (8) can be written as follows:

\[ \rho(\xi_{t-1}, \psi_{t-1}, v_{t-1}) \propto \rho(\xi_t^f, v_t, \psi_t) \rho(\xi_t, \psi_t - \xi_{t-1}, \psi_{t-1}) \times \rho(\xi_{t-1}, \psi_{t-1} - \xi_t, \psi_t), \]  

where \( \rho(\xi_t^f, v_t, \psi_t) = \rho(\xi_t^f, \psi_t) \rho(v_t, \psi_t). \)

It is important to note that this approach may require tuning the barrier function parameters to achieve good estimation performance. However, with proper implementation, it is an effective method for handling inequality constraints in particle forward filtering and backward smoothing. The particle filter designed in Eq. (11) can be written as follows:

\[ w_{t}^{(i)} = \frac{\rho(\xi_t^f, v_t, \psi_t)}{\sum_{i=1}^{N} \rho(\xi_t^f, v_t, \psi_t) \rho(v_t, \psi_t)}, \]  

where \( w_{t}^{(i)} \) is the weight of the \( i \)-th particle at time \( t \). Eq. (13) will remain the same for the backward smoothing because the virtual function does not influence this. The particle filtering approach for NMPC, augmented with a barrier function given in Eq. (17), integrates the treatment of inequality constraints. Consequently, the resulting NMPC framework is referred to as the improved NMPC.

3 Simulation and results

A nonlinear and highly coupled dynamical model of a robot vehicle is presented in this section. The dynamical model is then converted to a discrete-time model, and the proposed NMPC approach using particle filtering is applied. The results of NMPC with inequality constraints are compared with those of NMPC with equality constraints.

3.1 Dynamical model of a robot vehicle

The dynamical representation of the velocity vectors in the x-axis and y-axis are given as follows:

\[ \dot{x} = V \cos(\mu + \phi), \]
\[ \dot{y} = V \sin(\mu + \phi), \]  

where \( V \) is the velocity of the robot vehicle, which makes an angle \( \phi \) called the slip angle with the robot vehicle as shown in Figure 1, and \( \mu \) is the direction angle of the robot vehicle. The angular velocity of the robot vehicle is given by:

\[ \dot{\mu} = \frac{V}{R}, \]  

where \( R \) is the radius of the robot path and is given by the length OC, which is perpendicular to the velocity vector \( V \), as shown in Figure 1. To draw the expression for the radius \( R \) in Figure 1, we apply the sine rule to the OCA and OCB triangles, respectively, as follows:

\[ \frac{\sin(\phi_x - \phi)}{r_f} = \frac{\sin\left(\frac{\pi}{2} - \phi_x\right)}{R}, \]  

\[ \frac{\sin(\phi - \phi_b)}{r_b} = \frac{\sin\left(\frac{\pi}{2} - \phi_b\right)}{R}. \]  

Figure 1: Dynamical model of a robot vehicle.
Note that the net velocity vectors of the front and back wheels at point A and point B are in the direction of $\phi_f$ and $\phi_b$, respectively, and the instantaneous rolling center $O$ is the intersection of the lines OA and BO. Furthermore, $r_f$ and $r_b$ are the front and back distances from points A to C and points B to C, respectively. Eq. (22) can be rewritten as follows:

\[
\begin{align*}
\sin(\phi_f) \cos(\phi) - \sin(\phi) \cos(\phi_f) &= \frac{\cos(\phi_f)}{r_f}, \\
\sin(\phi) \cos(\phi_b) - \sin(\phi_b) \cos(\phi) &= \frac{\cos(\phi_b)}{r_b}.
\end{align*}
\]  

(23)

In our case, the net velocity vector of the back wheel is zero, i.e., $\phi_b = 0$. Furthermore, we can rewrite the following expression:

\[
\tan(\phi_f) \cos(\phi) - \sin(\phi) = \frac{r_f}{R},
\]

\[
\sin(\phi) = \frac{r_b}{R}.
\]  

(24)

Adding Eq. (24) and rearranging for $R$:

\[
R = \frac{r_f + r_b}{\tan(\phi_f) \cos(\phi)}.
\]  

(25)

The angular velocity of the robot vehicle is given by:

\[
\dot{\mu} = \frac{V}{r_f + r_b} \tan(\phi_f) \cos(\phi).
\]  

(26)

Next, we will drive the expression for the angle of slip $\phi$; for that, we will consider Eq. (24) and divide them as follows:

\[
\frac{\tan(\phi_f)}{\tan(\phi)} - 1 = \frac{r_f}{r_b}.
\]  

(27)

Rearranging the aforementioned equation results in the following expression:

\[
\phi = \tan^{-1} \left( \frac{r_b}{r_f + r_b} \frac{\tan(\phi_f)}{r_f} \right).
\]  

(28)

Overall, the dynamic model for the robot vehicle shown in Figure 1 is written as follows:

\[
\begin{align*}
\dot{x} &= V \cos(\mu + \phi), \\
\dot{y} &= V \sin(\mu + \phi), \\
\dot{\mu} &= \frac{V}{r_f + r_b} \tan(\phi_f) \cos(\phi), \\
\phi &= \tan^{-1} \left( \frac{r_b}{r_f + r_b} \frac{\tan(\phi_f)}{r_f} \right).
\end{align*}
\]  

(29)

### 3.2 Discrete-time model

The dynamical model given in Eq. (29) is the continuous time that needs to be converted to a discrete-time model in order to apply the proposed sampling-based motion planning algorithm. Samples are taken at each time step $\tau$ with the time interval $\delta \tau$. Eq. (29) can be written in the discrete-time domain as follows:

\[
\begin{align*}
x_{t+1} &= x_t + \delta \tau V \cos(\mu_t + \phi_t), \\
y_{t+1} &= y_t + \delta \tau V \sin(\mu_t + \phi_t), \\
V_{t+1} &= V_t + \delta \tau a_t, \\
\mu_{t+1} &= \mu_t + \frac{V_t}{r_f + r_b} \sin(\phi_t), \\
\phi_t &= \tan^{-1} \left( \frac{r_b}{r_f + r_b} \frac{\tan(\phi_f)}{r_f} \right),
\end{align*}
\]  

(30)

where $a_t$ is the discrete-time acceleration at time step $\tau$. The discrete-time control signals consist of discrete-time acceleration and front-wheel steering angle, respectively, and are given as follows:

\[
\psi_k = [a_k, \phi_k]^T.
\]  

(31)

The purpose of the proposed algorithm is to optimize control actions $u_k$ so that the desired track is followed by the robot vehicle.

### 3.3 Simulation results

The MATLAB environment was used for simulation. Figure 2 shows the desired reference track with upper and lower boundaries. The desired reference track is given by $d = 3 \cos(0.5\beta_t) + 3 \sin(0.2\beta_t)$, where $\beta_t$ is the interval of 0.5 with the total length of 50 as follows: $\delta_t = 0.5, 1, 1.5, \ldots, 50$. Furthermore, the upper and lower boundaries are given as $d_{\text{upper}} = d + 0.25$ and $d_{\text{lower}} = d - 0.25$, respectively.

The desired reference track was followed by the robot vehicle that has a discrete-time dynamical model given in Eq. (30) and the control signals in Eq. (31). The state vector is given by $x = [x_t, y_t, V_t, \mu_t]^T$ with initial values $\xi_0 = [0, 1, 8, 3, -\pi/4]^T$, respectively. There are four prediction horizons, one for each state given in state vector $\xi$, and the number of particles was taken 100 for simulation. The optimized states and control signals were derived using Eq. (15), whereas the importance weights $w_k^{(i)}$ were computed using Eq. (11) of particle forward filtering approach for NMPC and Eq. (19) of particle forward filtering approach for NMPC with inequality constraints. In both cases, particle backward smoothing was applied using the same
Figure 2: Reference track with upper and lower boundaries.

Figure 3: Comparison between improved-NMPC with barrier function to incorporate inequality constraints and normal-NMPC.

Figure 4: Comparison of errors between improved-NMPC and normal-NMPC.
Eq. (13). Figure 3 shows the optimized path for both cases, and results are computed. The blue line shows the track generated by the particle filtering approach for NMPC with a barrier function to incorporate the inequality constraints named improved-NMPC, whereas the green line shows the track generated by the particle filtering approach for NMPC without a barrier function named normal-NMPC.

Figure 4 displays the error signals comparing improved NMPC and normal NMPC. In order to evaluate the error signal for improved NMPC, two trajectories were considered: the reference trajectory and the trajectory designed by the improved NMPC. The error signal was obtained by computing the difference between these two trajectories. Similarly, for the normal NMPC, the error signal was evaluated by comparing the reference trajectory and the trajectory designed by the normal NMPC. Again, the error signal was computed as the discrepancy between these two trajectories. Figure 4 provides a visual representation of the error signals for both improved NMPC and normal NMPC, highlighting the differences between the reference trajectories and the trajectories generated by the respective control methods.

Furthermore, Figure 5 illustrates the control signals given in Eq. (31). It represents the acceleration and steering angle of the robot vehicle. The green signals show the control effort in normal NMPC, whereas blue signals show the control effort for improved NMPC with barrier function. It can be concluded that the control effort for improved NMPC is less than that of the normal NMPC, which shows the superiority of the proposed controller. Moreover, it can also be seen from Figure 3 that the improved NMPC performs efficiently to track the desired trajectory, whereas the normal NMPC is inefficient to track the trajectory.

4 Conclusion

This study presents an NMPC approach using particle filtering to solve a discrete-time dynamical model of a robot vehicle. Particle filtering is an algorithm that is used to implement Bayesian filtering for nonlinear and non-Gaussian systems. It is an SMC method that involves representing the posterior distribution using a set of weighted particles that are propagated through time using a recursive algorithm. The optimal control actions are estimated using this estimation from the reference signal with upper and lower boundaries. Furthermore, the particle forward filtering and backward smoothing approach is used to iteratively estimate the posterior probability distribution over the state space at each time step based on observed data and prior knowledge and use this estimate to predict future behavior and optimize the control actions. Moreover, a barrier function is introduced to penalize the likelihood of particles that violate the constraints in NMPC. Finally, the simulations are performed to verify the performance of the proposed NMPC with constraints, and it is concluded that NMPC with penalized function can efficiently track the desired reference trajectory.

Funding information: This research received no external funding.
Author contributions: The research idea, conceptualization, and methodology were formulated by M.A. and M.M.S. The controller design, software simulations, and validation were performed by M.A. Formal analysis and investigation were performed by M.M.S. Original draft was written by M.A. Writing – review and editing, and supervision by M.M.S. All authors have read and agreed to the published version of this manuscript.

Conflict of interest: The authors state that there is no potential conflict of interest.

References


