Gaussian Process-based Model Predictive Controller for Connected Vehicles with Uncertain Wireless Channel*

Hassan Jafarzadeh ¹ and Cody Fleming²

Abstract—In this paper, we present a data-driven Model Predictive Controller that leverages a Gaussian Process to generate optimal motion policies for connected autonomous vehicles in regions with uncertainty in the wireless channel. The communication channel between the vehicles of a platoon can be easily influenced by numerous factors, e.g., the surrounding environment, and the relative states of the connected vehicles, etc. In addition, the trajectories of the vehicles depend significantly on the motion policies of the preceding vehicle shared via the wireless channel and any delay can impact the safety and optimality of its performance. In the presented algorithm, Gaussian Process learns the wireless channel model and is involved in the Model Predictive Controller to generate a control sequence that not only minimizes the conventional motion costs, but also minimizes the estimated delay of the wireless channel in the future. This results in a farsighted controller that maximizes the amount of transferred information beyond the controller’s time horizon, which in turn guarantees the safety and optimality of the generated trajectories in the future. To decrease computational cost, the algorithm finds the reachable set from the current state and focuses on that region to minimize the size of the kernel matrix and related calculations. In addition, we present an efficient recursive approach to decrease the time complexity of developing the data-driven model and involving it in Model Predictive Control. We demonstrate the capability of the presented algorithm in a simulated scenario.

I. INTRODUCTION

Connected, autonomous vehicles (CAVs) have the potential to identify, collect, process, exchange, and transmit real-time data leading to greater awareness of – and response to – events, threats, and imminent hazards within the vehicle’s environment. At the core of this concept is a networked environment supporting very high speed interactions among vehicles (V2V), and between vehicles and infrastructure components (V2I) to enable numerous safety and mobility applications. When combined with automated vehicle-safety applications, vehicle connectivity provides the ability to respond and react in ways that drivers cannot, or cannot in a timely fashion, significantly increasing the effectiveness of crash prevention and mitigation. One application of CAVs involves Cooperative Adaptive Cruise Control (CACC), in which the following vehicle not only localizes the preceding vehicle via on-board sensing, but also receives information of the preceding vehicles’ current state(s) through a wireless channel [1]. CACC is an extension to Adaptive Cruise Control that is capable of applying acceleration/deceleration depending on the existence of a preceding vehicle in a certain range. ACC is one of the first concepts enabling automation to involve a safety concern (i.e. time-to-collision) in its policy.

Real-time control algorithms, e.g., Model Predictive Control (MPC), provide a foundation to significantly improve CACC performance. MPC involves generating a sequence of motion policies for some number of time steps in the future and executing a subset of this sequence in a receding fashion. Predicting the optimal trajectory into the future benefits the system in different ways, such as providing safety guarantees (e.g., obstacle avoidance) and minimizing control effort. Furthermore, in the context of CAVs, the predicted motion policy can be shared via a communication channel with the other vehicles. Being aware of future vehicle states allows better agility, maneuverability, safety, and optimality [2].

There is a growing literature on MPC in connected vehicle applications [3], [4]. However, most of this work assumes a perfect communication channel, meaning that an ego (following) vehicle receives packets from the preceding vehicle with no dropouts or delay. However, communication delays present a challenge, especially in vehicle-to-vehicle, or V2V, communication systems, which are characterized by a dynamic environment, high mobility, and low antenna heights/power on the transceivers (e.g., between vehicles and roadside units, telecommunications, etc) [5]. Communication delay can impact the performance of vehicle platoons in myriad ways. String stability is seriously compromised by the communication delay introduced by the network [6]. Liu et al assume a constant communication delay (i.e., a very benign delay) and propose a simple fix to preserve string stability that is not optimal. The sub-optimality of their approach comes from the fact that controller designs do not explicitly account for delay in the communication links. In practice a wireless channel inevitably contains random delays, and packet losses vary based on the surrounding environment. Therefore, an approach of multiple time-varying delays was considered in [7], which implements adaptive feedback gains to compensate for the errors originated from outdated packets, provides upper bound estimates for time delays, and examines stability conditions using Linear Matrix Inequality techniques.

However, little work has been done on the effect of uncertainty of communication channel on the control policy of CAVs, and to our knowledge none have considered the wireless network itself as a dynamic variable. Typically,
when the policy of a connected vehicle is optimized over a time horizon based on a given behaviour of the channel, it is assumed that the network will not vary by changing the state of the vehicle. However, the states of the communicating vehicles are important factors that influence the wireless channel.

States of the vehicle and wireless communication channel (e.g. PDR, or Packet Delivery Rate) are two interdependent variables, where changing either will affect the other [2]. The ego vehicle should not only take into account the cost of control effort and its longitudinal distance from the preceding vehicle, but also the state of the wireless network should be involved in its controller model. For instance, the state of the ego vehicle where the communication channel has zero PDR is not desirable because at this state the ego vehicle will not receive any packet from the lead vehicle which in turn results in a sequence of conservative controls (because of the safety concerns) that impacts the overall performance. Accordingly, having a more stable wireless channel with a higher PDR improves the overall performance of the CAVs; however the challenge is, being data-driven, the wireless communication channel cannot be formulated directly as a straightforward mathematical model to be merged into the model predictive controller.

To consider the PDR in MPC, we use Gaussian Processes (GP) which is a model-based method. GP is preferable in this work because the variations of the environment can be learned and used as a proxy model representing the behaviour of the wireless network for a given time horizon. Also, being a probabilistic method, the uncertainty can be incorporated into a long-term prediction (time horizon) by using GPs and the impact of the errors can be reduced [8], [9]. This is a data-efficient approach compared to the model-free methods (e.g. Q-learning or TD-learning) and can be used without additional interaction with the environment [10]. After obtaining the proxy model, it can be used in MPC to generate a motion policy in which the PDR, control effort and the highway capacity usage are balanced in an optimal way. To treat the problem without building a proxy model, DMPC (Data-and Model-Driven Predictive Control) can be applied, which is an iterative learning control approach. DMPC requires a feasible trajectory from an initial to terminal state, and iterates on the most recently generated full trajectory to find a better one until converging to the optimal trajectory [11].

GPs have been used to model the dynamics of different systems, for instance leveraging Gaussian Processes to model cart-pole system and a unicycle robot [12] and inverted pendulum [13]. We use the same approach of [10], which is an improvement to Probabilistic Inference for Learning Control (PILCO) methodology initially presented in [12] that has the ability to propagate uncertainty through the time horizon of a predictive controller and learn the parameters of a LTI system.

However, PILCO and its variants are computationally expensive, and in the context of CAVs it is not critical to learn a LTI system. Another challenge with the majority of learning algorithms is that they often require too many trials to learn. For example, learning the mountain-car tasks often requires hundreds or thousands of trials, independent of whether using policy or value iterations, or policy search methods [14], [15]. Thus the reinforcement learning algorithms have limited applicability to many real-world applications, e.g. mechanical systems, especially if their system dynamics change rapidly (such as wearing out quickly in low-cost robots) [12].

To decrease the size of the learned GP model in MPC, we use the concept of N-step reachable set to focus on a small part of the model required to find the optimal motion policy from the current state. Also, we exploit the repetitiveness of the MPC to develop a recursive approach to decrease the computational burden of the GP in MPC. Furthermore, our approach learns the parameters of a wireless channel and integrates this information with known or explicit dynamical models. Therefore, the contributions of this paper are:

1) a computationally efficient controller that leverages and extends state-of-the-art learning algorithms, and
2) a control architecture for CAVs that accounts for communication uncertainty in a locally optimal way.

II. BACKGROUND: MPC SCHEME FOR CAVS

Consider a platoon of autonomous vehicles that generate their own optimal motion policy and share it with other vehicles in the platoon through a wireless communication channel.

Given a leader-following pair of vehicles, let \( x_t \in \mathbb{R}^n \) and \( u_t \in \mathbb{R}^m \) be the state and control vectors of the ego, or following, vehicle at time step \( t \). Its dynamical system is given by:

\[
x_{t+1} = f(x_t, u_t).
\]

(1)

We assume that \( f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) is a smooth function. We first present the model predictive controller of the ego vehicle under the condition of perfect communication channel and then imperfect communication channel.

A. Perfect Communication Channel

In this case we assume that there is no delay in the wireless channel and the sent packet from the predecessor vehicle, that contains its predicted trajectory for \( N \) time steps in the future, is being received by the ego vehicle at no time. The controller of the ego vehicle for this case is proposed as an MPC architecture that solves the following finite-horizon optimization model with length of \( N \) at time \( t \):

\[
\min_{u_t} J = Q(x_{t+N\vert t}) + \sum_{k=t}^{t+N} h(x_{k\vert t}, u_{k\vert t}) \tag{2a}
\]

s.t. \( x_{k+1\vert t} = f(x_{k\vert t}, u_{k\vert t}) \)

(2b)

\( x_{k\vert t} = x_t \)

(2c)

\( \Phi(x_{k\vert t}, x^p_{k\vert t}) \geq \varphi \)

(2d)

\( x_{k\vert t} \in \mathcal{X}, \quad u_{k\vert t} \in \mathcal{U} \quad \forall k \in \{ t, \ldots , t+N \} \).  

(2e)

where \( h(x_{k\vert t}, u_{k\vert t}) \) is the stage cost and defined as:

\[
h(x_{k\vert t}, u_{k\vert t}) = \| x_{k\vert t} - x^p_{k\vert t} \|^2_{R_x} + \| x_{k\vert t} - x^ref_{k\vert t} \|^2_{R_s} + \| u_{k\vert t} \|^2_{R_u} \tag{3}
\]
and \( x_{k|t} \) and \( u_{k|t} \) are the state and control input vectors at step \( k \) predicted at time \( t \), respectively, which are shown as the solution of model (2):

\[
\begin{align*}
    x_t & = \left[ x_{t+1|t}, x_{t+2|t}, \ldots, x_{t+N|t} \right] \\
    u_t & = \left[ u_{t+1|t}, u_{t+2|t}, \ldots, u_{t+N-1|t} \right].
\end{align*}
\] (4)

The receding horizon control law applies the first control input \( u_{t|t} \) of \( u_t \) to shift the state of the system to \( x_{t+1|t} \) and the process is repeated again from \( t + 1 \). The cost function includes four terms to track: the state of predecessor vehicle \( x^p_{k|t} \), a desired reference trajectory \( x^r_{k|t} \), minimal control effort, and minimum terminal cost \( Q \). The tuning positive (semi)definite matrices are defined by \( R_1, R_2, \) and \( R_3 \). The optimal state vector of the lead vehicle, \( x^p_{k|t} \), is sent via the wireless channel in the following packet

\[
x^p_t = \left[ x^p_{t+1|t}, x^p_{t+2|t}, \ldots, x^p_{t+N|t} \right].
\] (5)

The dynamical system and the initial condition are represented by (2b) and (2c). Constraint (2d) enforces the safety conditions which can be defined simply as a minimum euclidean distance or as the Time To Collision factor that cannot be less than a given parameter \( \varphi \)

\[
\Phi(x_{k|t}, x^p_{k|t}) = \begin{cases} \infty & v^p_{k|t} \leq v_{k|t} \\ \frac{d_{k|t}}{v^p_{k|t} - v_{k|t}} & v^p_{k|t} > v_{k|t}, \end{cases}
\] (6)

\( v_{k|t} \) and \( d_{k|t} \) are the velocity of the ego vehicle and its euclidean distance from its predecessor at time step \( k \) predicted at time \( t \). It is assumed that there is a feasible trajectory from the initial state to the terminal state at time step \( t \).

In this scheme (2), it is assumed that the communication channel is perfect (common in literature, e.g. [16]), which means that the trajectory of the predecessor vehicle \( x^p_{k|t} \), \( \forall k \in \{1, \ldots, N + t\} \) is transferred without any delay to the ego vehicle at the beginning of each time step, \( t \). In this paper, this assumption will be relaxed to capture the uncertainty of the wireless communication channel.

B. Imperfect Communication Channel

The communication channel can be impacted by numerous factors such as relative states of the connected vehicles, surrounding infrastructure and vehicles, free space disturbances, etc. We define the predicted state vector of the communication channel established between the ego vehicle and its predecessor as

\[
w_t = \left[ w_{t+1|t}, w_{t+2|t}, \ldots, w_{t+N|t} \right].
\] (7)

\( w_{t+k|t} \) shows the prediction for the packet delivery time of time step \( k + t \) between two vehicles calculated at time \( t \) (i.e. the inverse of packet delivery rate, PDR). The question that we address in this section is “How does predicted delivery time affect the policy of the ego vehicle, and how can this delivery time be involved in the optimization model of the controller?” If the estimated delivery time \( w_{t+k|t} \) is greater than the length of the time step used for the discretized dynamics in the general MPC formulation, there will be an estimated delay at time step \( t+k \). In this case, the ego vehicle should use the most recent available packet, at time \( t+k-1 \), sent from the lead vehicle.

Due to the uncertainty in the delivery time of the packets, the ego vehicle can receive several packets at a single time step, \( \{x^p_t, x^p_{t+1}, \ldots, x^p_{t+N} \} \), where \( t_1 < t_2 < \ldots < t_m \leq t + k \). Because the packet containing \( x^p_{t_m} \) is the most up-to-date (though not necessarily current at time \( t+k \)), it is used to calculate the motion policy of the ego vehicle.

\[
x^p_{t_m} = [x^p_{t_m|t}, x^p_{t_m+1|t}, \ldots, x^p_{t_m+N|t}].
\] (8)

Nevertheless, at time step \( t+k \), the first \( t+k - t_m \) states in this packet belong to the past, are not useful anymore and should be neglected. Therefore, the useful packet shrinks to:

\[
x^p_{t_m} = [x^p_{t+k|t}, x^p_{t+k+1|t}, \ldots, x^p_{t_m+N|t}].
\] (9)

The length of this packet determines the applicable time horizon in the model predictive controller that the ego vehicle can consider to involve the trajectory of the lead vehicle:

\[
N_{t+k} = t_m + N - (t+k).
\] (10)

where \( N_{t+k} \) denotes the length of time horizon at time \( t+k \), \( N_{t+k} \leq N \). Increases in the value of \( w_{t+k|t} \) decreases \( t_m \), which in turn decreases the length of the useful time horizon \( N_{t+k} \). Fewer useful states from the lead vehicle causes the ego vehicle to take more conservative policies to satisfy the safety constraints, and this approach results in more cost for the entire whole trajectory; i.e. the controller might find local optima due to lack of longer-term information.

In this work we add awareness of the cost to physical system performance due to communication delays, which updates the cost function of model (2) to:

\[
\min_{w_t} J = Q(x_{t+N|t}) + \sum_{k=t}^{t+N} [h(x_{k|t}, u_{k|t}) + y(w_{k|t})],
\] (11)

in which \( y \) is a positive definite function. Adding a notion of the packet delivery time of the wireless channel to the cost function of the model equips the system with a farsighted controller to guarantee the availability of the packet with a maximum possible length at each time step in the future, ultimately resulting in a safer and more optimal motion policy. Therefore, the objective function includes the cost of the delay in communication channel for the trajectories that result in a low quality state of communication channel, \( \omega_{t+k|t} \). Thus, the optimality condition forces the model to generate trajectories with desirable quality of the communication while satisfying the constraints. In the next section we use Gaussian Processes to estimate the state of wireless channel at each time step, \( w_{k|t} \), as a function of the states of two connected vehicles.

III. GAUSSIAN PROCESS FOR CAVs

In the following, we will present the main components of Gaussian Processes for connected autonomous vehicles and show how to involve it in model predictive control. Later in the section, an efficient recursive approach will be developed to improve the time complexity of the presented algorithm.
A. Probabilistic Model for Wireless Channel

Assume that the delay of packet delivery from the lead vehicle to the ego vehicle at time $t$ can be described by the following unknown function

$$\omega_t = \Omega(x_t, x_t^P, e) + \varepsilon$$  \hspace{1cm} (12)

where $e$ show the external environment and $\varepsilon$ is noise $\varepsilon \sim N(0, \sigma_e^2)$. In this paper, we consider a Gaussian Process setting where we seek deterministic control inputs $u_t$ that minimize the expected long-term cost

$$\min_{u_t} \left\{ J_{t \to t+N} + \sum_{k=t}^{t+N} y(\mathbb{E}_{\omega_{k|t}}[\omega_{k|t}]) \right\}. \hspace{1cm} (13)$$

$J_{t \to t+N}$ denotes the conventional cost function in MPC, i.e. equation (2a), and $y(\mathbb{E}_{\omega_{k|t}}[\omega_{k|t}])$ denotes the cost of expected delay in packet delivery at time step $k$ calculated at time $t$. To implement the GP we use $x = [x, x^P]^T, x \in \mathbb{R}^{2n}$ as training inputs. $x$ and $x^P$ indicate the state of ego vehicle and the state of lead vehicle associated with $x$, and $\omega \in \mathbb{R}^0$ as training target. A GP as a probabilistic, non-parametric model can be fully specified by a mean function $m(\cdot)$ and a covariance function $k(\cdot, \cdot)$ which is defined as squared exponential (Radial Basis Function, RBF):

$$k(x, x') = \sigma_0^2 \exp \left( -\frac{1}{2} (x-x')^T L^{-1} (x-x') \right). \hspace{1cm} (14)$$

where $\sigma_0^2$ is signal variance, and $L = diag([l_1^2, \ldots, l_{2n}^2])$ with length-scales $l_1, \ldots, l_{2n}$.  

Assuming that there are $r$ training inputs and corresponding training targets, we collect $X = [x_1, \ldots, x_r]^T$ and $\omega = [\omega_1, \ldots, \omega_r]^T$. Given the test input denoted $x_*$, the posterior predictive distribution of $\omega_*$ is Gaussian $p(\omega_*|x_*, \omega) = N(\omega_*|m(x_*), \sigma^2(x_*))$ where

$$m(x_*) = k(x, x_*)^T (K + \sigma_0^2 I)^{-1} \omega,$$

$$\sigma^2(x_*) = k(x, x_*) - k(x, x_*)^T (K + \sigma_0^2 I)^{-1} k(x, x_*). \hspace{1cm} (15)$$

$K$ is the Gram matrix with entries of $K_{i,j} = k(x_i, x_j)$ [17]. It is assumed that $p(x_{k|t} | x_{k|t}, \mu_k, \Sigma_k) = N(x_{k|t} | \mu_k, \Sigma_k)$, where $\mu_k$ and $\Sigma_k$ are the mean and covariance of $x_{k|t}$. From equation (13), the cost function (13) can be rewritten as

$$\min_{u_t} \left\{ J_{t \to t+N} + \sum_{k=t}^{t+N} y(\mathbb{E}_{\omega_{k|t}}[\omega_{k|t}]) \right\} \hspace{1cm} (16)$$

where $x_{k|t}$ is the aggregated vector including the state vectors of the ego and lead vehicles. The first $n$ elements of this vector (i.e. test input) in $x = [x, x^P]^T$ belongs to the ego vehicle and will be decided by the model predictive controller such that the overall cost takes a minimum value. The constraints (2b)–(2e) hold for this cost function. After finding the best control input vector $u_t$, the first control action, $u_{t|t}$, is applied and the state of the ego vehicle is updated to $x_{t+1|t}$.

B. Efficient Recursive GP-MPC

The GP-MPC algorithm needs to invert the Gram matrix, $K$, in cost function (17), which has time complexity of $O(r^3)$, where $r$ is the number of training data. For large training sets (ten thousands or more) construction of GP regression becomes an intractable problem. We improve the time complexity of the algorithm in this section, leveraging the concept of Reachable Set to decrease the size of matrix $K$.

Definition 1 (one-step reachable set $\mathcal{B}$): For the system $\{\}$, the one-step reachable set from the set $\mathcal{B}$ is denoted as

$$\text{Reach}(\mathcal{B}) = \left\{ x \in \mathbb{R}^n : \exists x(0) \in \mathcal{B}, \exists u(0) \in \mathcal{U} \right\}$$

s.t. $x = f(x(0), u(0))$ \hspace{1cm} (18)

Reach($\mathcal{B}$) is the set of states that can be reached in one time step from state $x(0)$. $N$-step reachable set are defined by iterating Reach($\cdot$) computations [18].

Definition 2 (N-step reachable set $\mathcal{R}_N(\mathcal{X}_0)$): For a given initial set $\mathcal{X}_0 \subseteq \mathcal{X}$, the $N$-step reachable set $\mathcal{R}_N(\mathcal{X}_0)$ of the system $\{\}$ subject to constraints (2d) and (2e) is defined as [18]:

$$\mathcal{R}_{t+1}(\mathcal{X}_0) = \text{Reach}(\mathcal{R}_t(\mathcal{X}_0)), \quad \mathcal{R}_0(\mathcal{X}_0) = \mathcal{X}_0, \quad t = \{0, \ldots, N-1\} \hspace{1cm} (19)$$

Using the $N$-step Reachable Set concept, we define sub-matrix $\tilde{K}(t)$ that is extracted from matrix $K$.

$$\tilde{K}_{i,j}(t) = \{ k(x_i, x_j)| x_i, x_j \in \mathcal{R}_N(x_i) \cap \mathcal{X} \}. \hspace{1cm} (20)$$

This will result in the following cost function

$$\min_{u_t} \left\{ J_{t \to t+N} + \sum_{k=t}^{t+N} y(\mathbb{E}_{\tilde{X}_{k|t}}[\tilde{X}_{k|t}])^T (\tilde{K}(t) + \sigma_0^2 I)^{-1} \tilde{\omega}_t \right\}, \hspace{1cm} (21)$$

where $\tilde{X}_t = \{ x \in \mathcal{R}_N(x_i) \cap \mathcal{X} \}$ and $\tilde{\omega}_t$ is the associated training output. Assume that the sampling has been executed randomly and for each time step, $\nu$ samples are available on average. The number of training data extracted from the overall training data, $\mathcal{X}$, would be $\nu N$, where $\nu N \ll r$. Constructing the sub-matrix $\tilde{K}(t)$ is straightforward and can be implemented based on the state vector of lead vehicle, $x_0^P$, and the defined safety constraint (2a).

Lemma 1 Given Matrix $\tilde{K}(t-1)$ denoting the sub-matrix extracted from matrix $K$ and containing $N$-step Reachable Set at time step $t - 1$, and its inverse $\tilde{K}^{-1}(t-1)$, then $\tilde{K}^{-1}(t)$ can be calculated in $O(\nu^2 N^2)$ time.

Proof. In the given matrix $\tilde{K}(t-1)$, $\nu$ training data representing the wireless channel at time step $t - 1$ should be removed because the controller has implemented one step of control input, $u_{t-1}$. In addition, training data representing time step $t + N$ that is obtained from $\mathcal{R}_N(x_k)$ should be added to the matrix. The result of these two steps will be matrix $\tilde{K}(t)$. According to our assumptions, $\nu$ training data will be removed and added, which keep the size of matrix $\tilde{K}^{-1}(t)$ the same, $\nu N$. 

We use the Sherman–Morrison formula [19] to find $K_{1,1}^{-1}(t-1)$ (i.e. matrix $K^{-1}(t-1)$ that its first column and row are removed) from $K^{-1}(t-1)$ remove the first column and the first row from matrix $K(t-1)$. Based on Sherman–Morrison formula

$$K_{1,1}^{-1}(t-1) = (K(t-1) - pq^T)^{-1}, \tag{22}$$

where $p$ and $q$ are defined as: $p = K_1(t-1) - e_1$, and $q = e_1$, $(e_i$ is $i^{th}$ canonical column vector). The right hand side of equation (22) can be expanded as:

$$K_{1,1}^{-1}(t-1) = K^{-1}(t-1) + \frac{K^{-1}(t-1)pq^TK^{-1}(t-1)}{1 - q^T K^{-1}(t-1)p} \tag{23}$$

$1 - q^T K^{-1}(t-1)p$ is assumed to be invertible. This term can be calculated in $O(\nu^2N^3)$, which comes from multiplication of the square matrix $K^{-1}(t-1)$ with dimension $(\nu N \times \nu N)$ and vector $p$ and $q^T$. We denote the new training set as $\tilde{X}_{t-1}$, that has one less training data than $X_{t-1}$.

In the second step, a new training data, $x$, is added.

$$M := \begin{bmatrix} \tilde{K}_{1,1}(t-1) & k(\tilde{X}_{t-1}, x) \\ k(\tilde{X}_{t-1}, x)^T & k(x, x) \end{bmatrix} \tag{24}$$

Now, given $K_{1,1}^{-1}(t-1)$, we can find the inverse of matrix $M$ by using Schur complement [20]. The Schur’s complement of $K_{1,1}(t-1)$ in matrix $M$ is given by:

$$M/K_{1,1}(t-1) := k(x, x) - k(X_{t-1}, x)^T K_{1,1}^{-1}(t-1) k(X_{t-1}, x) \tag{25}$$

Assuming that $M/K_{1,1}(t-1)$ is invertible, $M^{-1}$ can be calculated in $O(\nu^2N^2)$ as:

$$M_{2,2}^{-1} = (M/K_{1,1}(t-1))^{-1} \tag{26}$$

$$M_{2,1}^{-1} = -M_{2,2}^{-1}k(\tilde{X}_{t-1}, x)^TK_{1,1}^{-1}(t-1)$$

$$M_{1,2}^{-1} = -K_{1,1}^{-1}(t-1)k(\tilde{X}_{t-1}, x)M_{2,2}^{-1}$$

$$M_{1,1}^{-1} = K_{1,1}^{-1}(t-1) - M_{1,2}^{-1}k(\tilde{X}_{t-1}, x)^TK_{1,1}^{-1}(t-1) \tag{27}$$

Removing and adding a row and a column both are calculated in $O(\nu^2N^2)$. These two steps should be repeated for $\nu$ times, which yields to time complexity of $O(\nu^3N^2)$.

Details of the algorithm are presented in Algorithm 1.

**Algorithm 1 : Calculating $K^{-1}(t)$ from $K^{-1}(t-1)$**

1: load training data $X$, $K(t-1)$ and $K^{-1}(t-1)$
2: $\tilde{X}_t \leftarrow \mathcal{R}_N(x_t) \cap X$
3: $X_{t-1} \leftarrow X_{t-1}$
4: calculate $K(t)$, equation (20)
5: $E \leftarrow K(t-1), E^{-1} \leftarrow K^{-1}(t-1)$
6: for $i = 1 \to \nu$
7: $p \leftarrow E - e_1$, and $q = e_1$
8: $E_{1,i} \leftarrow E - pq^T K^{-1}(t-1)$, equations (22), (23)
9: $X_{t-1} \leftarrow X_{t-1} - X_{t-1} \{1\}$
10: load new training data $x$ from $X$
11: $E \leftarrow \begin{bmatrix} E_{1,i} & k(\tilde{X}_{t-1}, x) \\ k(\tilde{X}_{t-1}, x)^T & k(x, x) \end{bmatrix}$
12: calculate $k(\tilde{X}_{t-1}, x)$ and $k(x, x)$
13: $S \leftarrow k(x, x) - k(\tilde{X}_{t-1}, x)^TpE_{1,i}^{-1}k(\tilde{X}_{t-1}, x)$, equation (25)
14: $M_{2,2}^{-1} \leftarrow S^{-1}$
15: $M_{2,1}^{-1} \leftarrow -M_{2,2}^{-1}k(\tilde{X}_{t-1}, x)^TE_{1,i}^{-1}$
16: $M_{1,2}^{-1} \leftarrow -E_{1,i}^{-1}k(\tilde{X}_{t-1}, x)M_{2,2}^{-1}$
17: $M_{1,1}^{-1} \leftarrow E_{1,i}^{-1} - M_{1,2}^{-1}k(\tilde{X}_{t-1}, x)^TE_{1,i}^{-1}$
18: $E^{-1} \leftarrow \begin{bmatrix} M_{1,1}^{-1} & M_{1,2}^{-1} \\ M_{2,1}^{-1} & M_{2,2}^{-1} \end{bmatrix}$
19: $\tilde{X}_{t-1} \leftarrow \tilde{X}_{t-1} \cup \{x\}$
20: end for
21: $K^{-1}(t) \leftarrow E^{-1}$
22: $\tilde{K}(t) \leftarrow E$
23: $\tilde{X}_t \leftarrow \tilde{X}_{t-1}$

for state $x_t$, $\tilde{X}_t$, the inverse of associated kernel matrix $K^{-1}(t)$ is calculated according to Lemma 1. Finally, the cost function (21) is built, and the GP-MPC proceeds by solving a sequential quadratic program.

It is assumed that the lead vehicle has a constant velocity $5m/s$ and the ego vehicle should keep the safe distance of at least $\varphi = 2m$ form the lead vehicle, $\Theta(x_t|t), x_t^\nu(\varphi) \geq 2$, which is enforced by constraint (23). The time horizon is considered to be $N = 10$ and the length of each time step is $dt = 1s$. The upper and lower bound values of the state and input values are $\bar{x}_t \in [3, 10]$ and $\bar{x}_t \in [-3, 2]$.

The trajectory of the lead vehicle is depicted by a red line in Figure 1, and we would like to design a controller that finds an optimal trajectory for the ego vehicle that not only minimizes the conventional motion cost, but also minimizes the expected delay time in the packet delivery (recalling again that the controller knows nothing about the contours in Figure 1 in advance). The blue line shows the optimal trajectory generated by GP-MPC for the ego vehicle. It can be seen from the second Figure 2 that, to maintain adequate wireless channel status (and to optimally balance control cost, safety, wireless performance, and do this as fast as possible), the ego vehicle decelerates to increases its distance from the lead vehicle. This policy sacrifices a bit of short-term performance in order to maintain connectivity,
Fig. 1: Optimal trajectory of ego vehicle generated by GP-MPC to avoid a region with a considerable delay time in packet delivery.

Fig. 2: Optimal control input $\ddot{x}$, state $\dot{x}$ and inter-vehicular distance $d_v$ over the time horizon.

improving situational awareness and ultimately long-term performance.

V. CONCLUSIONS

In this paper, an efficient Model Predictive Control algorithm based on Gaussian Processes is presented to account for uncertainty in the communication channel of motion policies in connected vehicle applications. The presented algorithm learns the wireless channel model in terms of expected packet delivery time by leveraging Gaussian Processes. Due to the substantial amount of available training data, the obtained GP model is very large and computationally expensive to deal with.

To solve this problem, the algorithm focuses on the $N$-step reachable set from the current state of the ego vehicle as the useful training data set. This decreases the size of the required training data for the current state of the vehicle dramatically. Subsequently the resultant kernel matrix would be substantially smaller than the original Gram matrix, which needs less computational effort to be inverted and multiplied. In addition, because a major part of the computation involved in GP is conducted to find the inverse of the kernel matrix, the controller exploits the recently calculated and readily available inverse of the kernel matrix from the previous state. We use Sherman-Morrison formula and Schur complement to find the inverse of the current kernel matrix after updating the training data set. These steps decrease the running time to find the expected packet delivery time of the wireless channel for the given time horizon $N$ from $O(n^3)$ to $O(\nu^2N^2)$ where $n$ and $\nu$ are the number of overall training data and drawn data for a single time step, respectively, and $\nu N \ll n$. To demonstrate the approach, a simulation of a leader-follower scenario for two connected autonomous vehicles is developed and the results of the algorithm presented.

REFERENCES

[1] P. Varaiya, “Smart cars on smart roads: problems of control,” IEEE Transactions on automatic control, vol. 38, no. 2, pp. 195–207, 1993.
[2] H. Jafarzadeh and C. Fleming, “Learning model predictive control for connected autonomous vehicles,” in 2019 IEEE 58th Conference on Decision and Control (CDC). IEEE, 2019, pp. 2336–2343.
[3] R. Schmied, H. Waschl, R. Quirynen, M. Diehl, and L. del Re, “Nonlinear mpc for emission efficient cooperative adaptive cruise control,” IFAC-promotion online, vol. 48, no. 23, pp. 160–165, 2015.
[4] T. Stanger and L. del Re, “A model predictive cooperative adaptive cruise control approach,” in 2013 American Control Conference. IEEE, 2013, pp. 1374–1379.
[5] W. Viriyasitavat, M. Boban, H.-M. Tsai, and A. Vasilakos, “Vehicular communications: Survey and challenges of channel and propagation models,” IEEE Vehicular Technology Magazine, vol. 10, no. 2, pp. 55–66, 2015.
[6] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, “Effects of communication delay on string stability in vehicle platoons,” in IFSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585). IEEE, 2001, pp. 625–630.
[7] A. Petriollo, A. Salvi, S. Santini, and A. S. Valente, “Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays,” Transportation research part C: emerging technologies, vol. 86, pp. 372–392, 2018.
[8] J. G. Schneider, “Exploiting model uncertainty estimates for safe dynamic control learning,” in Advances in neural information processing systems, 1997, pp. 1047–1053.
[9] J. Kocijan, Modelling and control of dynamic systems using Gaussian process models. Springer, 2016.
[10] S. Kamthe and M. P. Deisenroth, “Data-efficient reinforcement learning with probabilistic model predictive control,” arXiv preprint arXiv:1706.06491, 2017.
[11] H. Jafarzadeh and C. Fleming, “Dmpc: A data-and model-driven approach to predictive control,” Automationa, vol. 131, p. 109729, 2021.
[12] M. Deisenroth and C. E. Rasmussen, “Pilco: A model-based and data-efficient approach to policy search,” in Proceedings of the 28th International Conference on machine learning (ICML-11), 2011, pp. 465–472.
[13] M. Cutler and J. P. How, “Efficient reinforcement learning for robots using informative simulated priors,” in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 2605–2612.
[14] A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, and S. Levine, “Collective robot reinforcement learning with distributed asynchronous guided policy search,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 79–86.
[15] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.
[16] R. Firoozi, S. Nazari, J. Guanetti, R. O’Gorman, and F. Borrelli, “Safe adaptive cruise control with road grade preview and v2v communication,” arXiv preprint arXiv:1810.09000, 2018.
[17] C. E. Rasmussen, “Gaussian processes in machine learning,” in Summer School on Machine Learning. Springer, 2003, pp. 63–71.
[18] F. Borrelli, A. Bemporad, and M. Morari, Predictive control for linear and hybrid systems. Cambridge University Press, 2017.
[19] E. Juárez-Ruiz, R. Cortés-Maldonado, and F. Pérez-Rodríguez, “Relationship between the inverses of a matrix and a submatrix,” Computation y Sistemas, vol. 20, no. 2, pp. 251–262, 2016.
[20] F. Zhang, The Schur complement and its applications. Springer Science & Business Media, 2006, vol. 4.