Benedittis, G. D.; Wymeersch, H.; Charalambous, T.

**Intersection Crossing of Autonomous Vehicles for Communication Links with Packet Losses**

**Published in:**
2022 IEEE 61st Conference on Decision and Control (CDC)

**DOI:**
10.1109/CDC51059.2022.9993266

Published: 10/01/2023

**Document Version**
Peer reviewed version

---

Please cite the original version:

---

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
Intersection Crossing of Autonomous Vehicles for Communication Links with Packet Losses

Giacomo De Benedittis, Henk Wymeersch, and Themistoklis Charalambous

Abstract—This work aims at addressing the problem of intersection crossing for autonomous vehicles in the presence of a lossy communication channel (that may cause the loss of data packets in the communication) between the vehicle and a central coordinator. This work aims to provide a solution that guarantees the crossing of the intersection, despite the packet losses. Our approach consists of an optimal control algorithm in which at every time step the optimal control sequence simulating a communication ensemble between the vehicle and the coordinator is computed. Since the model is stochastic and the channel imperfect, the intersection crossing condition is modeled as a chance constraint. Once the control sequence is found, the first step is applied and the optimization is performed again over a shrunk horizon. A real-time state observer using optimal Kalman filter observation gains is used. The performance of our approach is tested on a simplified dynamic car model. The sensitivity of the generated control sequence with respect to some key parameters (such as failure probability allowed and packet drop probability) is also considered.

Index Terms—Intersection crossing, autonomous vehicles, networked control systems, predictive control, shrinking horizon, chance constraints.

I. INTRODUCTION

The deployment of autonomous vehicles is expected to transform the way we perceive transportation nowadays, bringing several services and additionally improve the handling of persisting issues, such as road safety and traffic congestion, while at the same time improve fuel efficiency. One of the most critical areas during everyday driving is road intersections since coordination between different vehicles coming from and to different directions is required to avoid collisions. The coordination of multiple vehicles in this kind of situation can be achieved through different techniques, such as Vehicle-to-Vehicle decentralized communication or Vehicle-to-Infrastructure communications [1].

The problem of intersection crossing schedule is broken down into three parts: 1) determining an optimal priority list for vehicles waiting at the intersection or approaching it (usually treated through approximation [2] or heuristic methods [3], as most of the scheduling problems); 2) finding the optimal time interval for each vehicle to cross the intersection once the order is decided (solved with heuristic methods as well; see, for example, [4]), and 3) planning the optimal trajectory for each vehicle once the time slot has been decided. About the third part of the problem, it has already been treated in previous works for both continuous and discrete-time systems taking in account constraints for both physical limits of the single vehicle and coordination between different autonomous vehicles (AVs) in order to avoid collisions [5]–[8].

G. De Benedittis is an alumnus of the department of Mechanical Engineering of the Politecnico University of Milan, Italy. Actually he is an employee (internship) at Ferrari S.p.A., Italy in the Electric Motors design department. E-mail: giacomo.debenedittis@mail.polimi.it.

H. Wymeersch is with the Department of Electrical Engineering at Chalmers University of Technology, Sweden. E-mail: henkw@chalmers.se.

T. Charalambous is with the Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, Nicosia, Cyprus. He is also a Visiting Professor at the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland. E-mail: charalambous.themistoklis@ucy.ac.cy.

The present work takes a step back from the state-of-the-art algorithms already present in this field in order to address two factors that are often neglected in these trajectory planning problems: i) a stochastic dynamic system is used to take into account both process and measurement noises, and ii) the communication channel may not be perfect, therefore information can get lost at certain steps. The same problem has been treated in [9], and this work starts from there trying to improve the modelling of the interaction between the estimation error of the system and the resulting difference from the optimal trajectory that is computed at each time step, resulting in a Shrinking Horizon Stochastic Optimal Control problem. The way the stochasticity of the model is handled in terms of constraints is the one introduced by Okamoto et al. in [10], that uses chance constraints to steer at the same time the expected value and the covariance of the stochastic system within given bounds through a convex optimization problem, but in that case, the terminal region for the expected value is assumed to be known and set as the origin of the state space, while in this work also the optimal final state is determined by the solution of the optimization problem. This causes a variation in both the cost function and the constraints formulation. In [11], the problem of intersection crossing with an unreliable communication channel is first pointed out and solved by a chance-constrained optimization problem but the covariance is left free to evolve and the measurement noise is not taken into account. The application of optimal control to stochastic model has been extensively used for Stochastic Optimal Controllers and particularly in Stochastic Receding Horizon Control algorithms, also widely known as Stochastic Model Predictive Control (SMPC) [12], [13]. [14] applied SMPC to models with communication impairments, but in that case, the optimal trajectory is taken as determined and used as the origin of the states space at each time step: the state then is the deviation from the reference and has to be minimized (there is not an optimal trajectory computation). The problem of lossy networks and system uncertainty on MPC are considered separately in [15] and [16], respectively. Recently, Pezzuto et al. [17] addressed the possibility of using a MPC to deal with the problem of lossy networks having satisfying experimental results.

This work uses a discrete-time model of a vehicle moving straight with both process and measurement Gaussian noises and with the presence of an identically distributed probability of information loss between the vehicle and a central coordinator. To have an estimation of the state that takes into account the noises and the loss of information, a Kalman estimation method that takes into account intermittent communication formulated by Sinopoli et al. [18] is used. The intersection crossing is given as a chance constraint and the covariance is steered by a final covariance constraint. The problem is solved at every time step reducing the decision horizon in a Shrinking Horizon Constrained Optimization problem.

Notation. The sets of real numbers are denoted by $\mathbb{R}$, $\mathbb{R}^n$ is a vector space of real numbers with $n$ elements and $\mathbb{R}^{m \times n}$ is a matrix space of real numbers with $n$ rows and $m$ columns. The transpose of a vector $a$ is denoted by $a^T$ and $\sum_{i=a}^{b}$ is the sum of elements with
index $i$ going from value $a$ to $b$. By $\|\cdot\|_n$ the $n$–norm operator is meant. $E[\cdot]$ is the expected value of the argument $\cdot$ inside the brackets. $\text{tr}(\cdot)$ is the trace of a matrix, that is the sum of its diagonal elements. $0$ and $I$ are the matrix of zeros and the identity matrix, respectively, of appropriate dimensions.

The remainder of this paper is structured as follows. In Section II the system model, some preliminaries, and the problem setup are provided. In Section III we explain our approach and provide our algorithm. The performance of our algorithm is evaluated in Section IV. Finally, in Section V, we draw conclusions and discuss possible future directions.

II. SYSTEM MODEL AND PRELIMINARIES

The problem model from which this work starts is the trajectory planning for an autonomous vehicle crossing an intersection moving straight, considering process and measurement noises. The control of this trajectory planning is handled by a central coordinator. The communication channel between the vehicle and the central coordinator is not perfect, thus information packets given by the vehicle may get dropped with a probability $p_d$. The central coordinator receives the measurement data from the vehicle and having an initial estimate of state and error covariance of the car state it updates the estimation of the position using a Kalman filter. This model uses an observer without delay, where the estimate at a given time $k$ is based on the innovation error computed at that same time. The approach used for the solution of this problem is the application of a shrinking horizon Constrained Optimal Control to the stochastic dynamic model of the system, as treated in [9]. Therefore, the problem is solved as a constrained convex optimization problem of minimization of the cost function

$$J(x_0, \ldots, x_N, u_0, \ldots, u_{N-1}) =$$

$$E\left[\sum_{k=0}^{N} \left( (x_k - \hat{x}_k)^T Q_k (x_k - \hat{x}_k) + \sum_{k=0}^{N-1} u_k^T R_k u_k \right) \right]$$

(1)

where $x_k$ is the state of the state at every time step, $\hat{x}_k$ is the expected value of the state at every step as it will be computed by the optimization problem, $u_k$ is the system input at each time step. $Q \succeq 0$ and $R \succ 0$. This formulation is a modified version of the one used in [10], [19] and [20], with the difference that the objective of the optimization problem is not to bring the state vector to the origin of the space, but to generate an ideal state trajectory $\bar{x}_0, \ldots, \bar{x}_N$ and to keep the state $x_0, \ldots, x_N$ as near as possible to it.

A. Dynamic System definition

The model used, a discrete-time stochastic linear model of the car moving in straight direction, with both process and measurement noises, is given by

\[
x_{k+1} = A_x x_k + B_x u_k + D_x w_k, \quad \text{(2a)}
\]

\[
y_k = C_x x_k + G_x r_k, \quad \text{(2b)}
\]

where $x_k := [s_k \ v_k \ a_k]^T \in \mathbb{R}^{n_x}$ is the state vector at time $k$, containing the position, speed and acceleration of the vehicle, while $u_k \in \mathbb{R}^{n_u}$ is the system input and $y_k \in \mathbb{R}^{n_y}$ is the measured output. $w_k \in \mathbb{R}^{n_w}$ and $r_k \in \mathbb{R}^{n_r}$ are respectively process and measurement noise (both of them are assumed to be zero-mean Gaussian white noises with unit covariance). The matrices representing the discrete-time dynamic model, as defined in [21] are:

\[
A_x = A = \begin{bmatrix}
1 & h & 0 \\
0 & 1 & h \\
0 & 0 & 1 - \frac{h}{\tau}
\end{bmatrix}, \quad B_x = B = \begin{bmatrix}
0 \\
0 \\
\frac{h}{\tau}
\end{bmatrix}, \quad \text{(3)}
\]

where $h$ is the time discretization step of the model and $\tau$ is the acceleration-to-deceleration factor of the vehicle (therefore $n_x = 3$). In the system, all the components of the state are measured (i.e., $C_k = I \forall k$ and $n_y = 3$). The process and measurement noise matrices $D_x$ and $G_k$ are modelled depending on the case. $x_0$ is a Gaussian random variable (R.V.) with $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$. It is assumed that also the state at step $N$ is a Gaussian R.V., i.e., $x_N (x_N \sim \mathcal{N}(\mu_N, \Sigma_N))$. Even if $C = I$, due to the presence of noise, the measurement $y_k$ is noisy and hence, a Kalman filter is deployed (see Section II-B).

B. Kalman filter and communication model

The general formulation for a state observer is

\[
\begin{align*}
\hat{x}_{k+1|k} &= A \hat{x}_{k|k} + B u_k, \quad \text{(4a)}

\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + F_{k+1} (y_{k+1} - C \hat{x}_{k+1|k}), \quad \text{(4b)}
\end{align*}
\]

where $\hat{x}_{k+1|k}$ is the a priori estimation of the state at time $k + 1$, and $\hat{x}_{k+1|k+1}$ is the a posteriori estimation. The model adopted in this paper for the observer is the one formulated in [18], which is a Kalman filter with intermittent communication with the measurement system. The purpose of the Kalman filter is to minimize covariance of the error between the state and the estimation, i.e.,

$$e_k := x_{k+1} - \hat{x}_{k+1|k+1}. \quad \text{(5)}$$

The measurement error has zero mean at every time step and covariance that is assumed to be known at initialization (i.e., $e_0 \sim \mathcal{N}(0, \Sigma_0)$). $F_k$ is the gain of the observer, which gives a correction factor that is as high as the innovation error (the difference between the output estimation and the output measurement) is, and in the case of the Kalman filter with intermittent communication it is defined by the following equations:

\[
\begin{align*}
\Sigma_{k+1|k} &= A \Sigma_{k|k} A^T + D_x D_x^T, \quad \text{(6a)}

F_{k+1} &= \Sigma_{k+1|k} (G_{k+1} C_{k+1|k+1} + \Sigma_{k+1|k+1})^{-1}, \quad \text{(6b)}

\Sigma_{k+1|k+1} &= (I - \gamma_k F_{k+1}) \Sigma_{k+1|k}, \quad \text{(6c)}
\end{align*}
\]

where $\Sigma_{k+1|k}$ and $\Sigma_{k+1|k+1}$ are respectively the a priori and the a posteriori covariances at state $k + 1$. The variable $\gamma_k$ represents the information arrival to the coordinator, and starting from time 0 up to the end of the horizon, a sequence $\{\gamma_k\}$ is obtained in two different ways:

- Sampled distribution: $\gamma_k = \delta_k$, where $\delta$ is a binary series that is 0 with probability $p_d$ and 1 with probability $1 - p_d$.
- Direct probability distribution: $\gamma_k = 1 - p_d$ at every time step.

This approach has been used in [18] to provide deterministic upper and lower bounds for the covariance matrix evolution if the Kalman filter with intermittent communication is utilized for several time steps. This approach is used to see if a more deterministic approach can give meaningful results for this application.

By defining the error $e_k$ in (5), the output of the system $y_k$ can be written in a compact notation that includes both the cases of received or lost information from the vehicle (in the second case $y_{k+1} = C \hat{x}_{k+1|k} + C A \hat{x}_{k|k} + C B u_k$):

\[
y_{k+1} = (A x_k + B u_k + D_x w_k) + G_x r_{k+1} = C A x_k + C B u_k + G_x r_{k+1} + H_k \xi_k + C A e_k, \quad \text{(7)}
\]

where $H_k = [C D_k \ G_{k+1}]$ and $\xi_k = [w_k^T \ r_{k+1}^T]$. The state estimation at time $k + 1$ can then be written as:

\[
\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + F_{k+1} (y_{k+1} - C \hat{x}_{k+1|k}) = \hat{x}_{k+1|k} + \Delta_k \xi_k + \gamma_{k+1} F_{k+1} C A e_k, \quad \text{(8)}
\]
III. MAIN RESULTS

A. Estimation error evolution in presence of the Kalman filter

Using the formulation obtained in (2a) and (8), the recursive definition of $e_k$ can be obtained in the form

$$e_{k+1} = \Phi_{k+1}(Ae_k + Dw_k) - \gamma_k + F_{k+1}Gr_{k+1},$$

where $\Phi_k = (I - \gamma_k F_k C)$. By applying the recursion, it is possible to define a transition matrix $T(a, b) \in \mathbb{R}^{n_x \times n_x}$, such that

$$T(a, b) = (\Phi_k) (A\Phi_{b-1} \cdots (A\Phi_a), a, b \in \mathbb{N}^*, a \geq a).$$

With this formulation, knowing the noise covariance matrices, it is possible to know how the error evolves along the whole time horizon $(0, \ldots, N)$:

$$e_k = T(1, k)Ae_0 + \bar{T}_k \xi,$$

where $\xi = \begin{bmatrix} \xi_0^T & \xi_1^T & \cdots & \xi_{N-1}^T \end{bmatrix}^T \in \mathbb{R}^{Nn_\xi}$,

$$\bar{T}_k = \begin{bmatrix} [T(1, k)D & -T(2, k)A\gamma_k F_1 G_1]_T & \cdots & [T(N, k)D & -T(k, k)A\gamma_k F_{k+1} G_k]_T \end{bmatrix} \in \mathbb{R}^{n_x \times Nn_\xi}. $$

Inserting (11) inside the evolution of the estimation as written in (8) we get that

$$\dot{x}_{k+1|k+1} = Ax_k + Bu_k + \Delta_k \xi + \gamma_k + F_{k+1}CAT(1, k)Ae_0 + \gamma_k + F_{k+1}CAT\bar{T}_k \xi,$$

with

$$e_k = [\hat{x}_0 - x_0, \hat{x}_1 - x_1, \ldots, \hat{x}_{N-1} - x_{N-1}]^T \in \mathbb{R}^{(N+1)n_x},$$

and $D_k = \begin{bmatrix} 0 & T_1^T & \cdots & T_{N-1}^T \end{bmatrix} \in \mathbb{R}^{(N+1)n_x \times n_x}$.

The cost function can be rewritten with respect to the expanded variables:

$$J(\chi, v) = \text{tr}(\dot{Q}\Sigma_\chi) + \text{tr}(\ddot{Q}\Sigma_\chi) + 2\text{tr}(\dot{Q}\Sigma_\chi) + \text{tr}(\Sigma_\chi),$$

where $\dot{\chi} = A\chi_0 + Bv$, $\ddot{\chi} = \text{bikdiag}(Q_1, \ldots, Q_N)$, $\tilde{R} = \text{bikdiag}(R_1, \ldots, R_N)$, $\Sigma_\chi = \text{diag}(\chi - \chi)(\chi - \chi)^T$, $\Sigma_\chi = \text{diag}(\Sigma_\chi)$, $\Sigma_\chi = \text{diag}(\Sigma_\chi)$. From these definitions it can be noticed that the first term of (18) does not depend on the control sequence and can be excluded from the optimizations.

C. Decision variable and covariance matrices definitions

The control is defined as a state-feedback function of the previous time instants in the shape $v_k = l_k [\hat{x}_0^T \hat{x}_1^T \ldots \hat{x}_{N-1}^T]$ (the estimation is a posteriori for the observer without delay, a priori for the observer with delay). Therefore the system needs an augmentation so that $v$ can be written as

$$v = L\hat{\chi},$$

where

$$L(\chi, v) = \text{diag}(\chi - \chi)(\chi - \chi)^T,$$

In this notation, $L \in \mathbb{R}^{(N+1)n_x \times (N+1)n_x}$.

In the augmented system notation also the covariance is

$$\Sigma_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In order not to have an inverse matrix as design variable, $K := L(I - BL)^{-1}$ is used, with $K = [K_1 \cdots K_N]$ having $K_1 \in \mathbb{R}^{n_x \times n_x}$ and $K_N \in \mathbb{R}^{n_x \times (N+1)n_x}$ being block lower triangular.
With this formulation the main system variables become
\[
\hat{x}(t) = (I + BK)(A\hat{x}_0 + D\xi + Ee_0),
\]
(20)
\[
v = K(A\hat{x}_0 + D\xi + Ee_0).
\]
(21)
\(K\) is the design variable in the optimization problem to minimize the cost function. Now that it is defined, all the terms and constraints of the model need to be written as a function of \(K\). The planned trajectory for the vehicle is given by
\[
\hat{x}(t) = (I + BK)A\hat{x}_0.
\]
(22)
The estimation error evolution can be represented in the augmented system as
\[
e = \begin{bmatrix} 0 & 0 \\ 0 & A_e \end{bmatrix}e_0 + \begin{bmatrix} 0 \\ D_e \end{bmatrix}\xi = A_e e_0 + D_e \xi,
\]
(23)
where \(A_e \in \mathbb{R}^{(N+2)nx \times 2nx}\) and \(D_e \in \mathbb{R}^{(N+2)nx \times n_\xi}\).

For the development of our results, we define the following covariance matrices:
\[
\Sigma_{\hat{x}} = \mathbb{E}[(x - \hat{x})(x - \hat{x})^T],
\]
(24)
\[
\Sigma_{\chi} = \mathbb{E}[(x - \hat{x})(x - \hat{x})^T],
\]
(25)
\[
\Sigma_{x\chi} = \mathbb{E}[(x - \hat{x})(x - \hat{x})^T],
\]
(26)
\[
\Sigma_{xx} = \mathbb{E}[(x - \hat{x})(x - \hat{x})^T].
\]
(27)
It can be deduced that \(\Sigma_{\hat{x}} = \Sigma_{\chi} + \Sigma_{x\chi} + \Sigma_{xx}\). The four terms composing the covariance of the state with respect to the planned trajectory are respectively the estimation error covariance, the covariance of the estimation with respect to the planned trajectory and their two cross-covariances. Substituting (23) into (25),
\[
\Sigma_{x} = \mathbb{E}[(x - \hat{x})(x - \hat{x})^T] = \mathbb{E}[ee^T]
\]
(28)

The result of this implementation is a symmetric and positive definite matrix whose block-diagonal elements are the covariances obtained initializing the covariance as \(\Sigma_0\) and implementing the Kalman algorithm as in (6a)-(6c). The covariance of the estimation with respect to the planned value can be rewritten using (20) and (22) into (26):
\[
\Sigma_{\hat{x}} = (I + BK)(DD^T + E\Sigma_0 E^T)(I + BK)^T.
\]
(29)
The central part of (29) is the open-loop covariance \(\Sigma_{op} = (DD^T + E\Sigma_0 E^T)\).

The covariance matrix is defined as
\[
\Sigma_{x\hat{x}} = (I + BK)\Sigma_{op}(I + BK)^T
\]
(30)
and as can be seen by this formulation, it can be modified by the control action. Finally, substituting (23), (20) and (22) into (27), the cross covariance can be rewritten as follows:
\[
\Sigma_{xx} = (D_e D^T + A_e \Sigma_0 E^T)(I + BK)^T.
\]
(31)

D. Cost function formulation

Now that the design variable is defined and all the components of the cost function are rewritten as a function of it, also the cost function itself can be rewritten. Specifically, substituting (29), (31) and (21) into (18), we obtain the following formulation of the cost function \(J(K)\):
\[
J(K) = \text{tr}((I + BK)^T Q(I + BK) + K^T R K)(DD^T + E\Sigma_0 E^T) + (K^T R K)(A\hat{x}_0 \hat{X}_0 A^T) + 2\text{tr}(Q(I + BK)(E\Sigma_0 A_e + DD_e)),
\]
(32)
where \(Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(N+2)nx \times (N+2)nx}\).

E. Constraints formulation

1) Chance constraint: The chance constraint for the intersection crossing having a lower bound for failure probability \(p_f\) is given by
\[
Pr(x_N \leq x_{\text{crit}}) \leq p_f,
\]
Knowing that the states are normally distributed, the equality becomes
\[
Pr(\alpha^T \hat{x} \leq \beta) = \Phi \left( \frac{\beta - \alpha^T \hat{x}}{\sqrt{\alpha^T (\Sigma_{\hat{x}} + \Sigma_{\chi} + \Sigma_{x\chi} + \Sigma_{xx}) \alpha^T}} \right) \leq p_f,
\]
where \(\Phi(\cdot)\) is the cumulative distribution function of a zero-mean unit variance Gaussian R.V. By substituting \(\Sigma_{\hat{x}}\) with the definition obtained in (30), and by defining \(M = \Sigma_{op}\) and \(W = \Sigma_{x}\), we can write the relation
\[
\alpha^T (\Sigma_{\chi} + \Sigma_{x\chi} + \Sigma_{xx}) = v^T v + u^T u + \bar{u}^T \bar{u} + \bar{v}^T \bar{v},
\]
(33)
where \(\bar{u} = M(I + BK)^T\bar{v}\) and \(\bar{v} = W\alpha\). Using \(\bar{u}\) and \(\bar{v}\) the chance constraint becomes:
\[
\beta - \alpha^T \hat{x} - (\bar{u}^T \bar{v}^T + \bar{u}^T \bar{u} + \bar{u}^T \bar{v} \leq 0.
\]
Since both \(\bar{u}\) and \(\bar{v}\) are vectors, then the term present in the chance constraints can be rewritten as a function of their 2-norms:
\[
\sqrt{\bar{u}^T \bar{u} + \bar{v}^T \bar{v} + 2\bar{u}^T \bar{v}} = ||\bar{u}|| + ||\bar{v}||
\]
(34)
so the chance constraint is strengthened1 if written in the form
\[
\beta - \alpha^T \hat{x} - (||\bar{u}|| + ||\bar{v}||)\Phi^{-1}(p_f) \leq 0.
\]
(35)
Then, substituting \(\bar{u}\) and \(\bar{v}\), (35) becomes the final form of the chance constraint given by
\[
\beta - \alpha^T \hat{x} - (||M(I + BK)^T\alpha|| + ||W\alpha||)\Phi^{-1}(p_f) \leq 0.
\]
(36)
2) Terminal covariance constraint: Since only \(\Sigma_{\hat{x}}\) can be modified by the control action, while \(\Sigma_{\chi}\) is determined only by the packet drop probability and the consequent \(\chi_k\) sequence, a final covariance constraint can only be formulated as
\[
E_N \Sigma_{\hat{x}} = E_N (I + BK)\Sigma_{op}(I + BK)^T E_N^T \leq \Sigma_{\text{lim}},
\]
with \(E_N = \begin{bmatrix} 0 & 0 & \ldots & 1 \end{bmatrix}^T \in \mathbb{R}^{n_x \times (N+2)nx}\). Under the assumption \(\Sigma_{\text{lim}} \geq 0\), the constraint can be rewritten as
\[
I - \frac{1}{4} (E_N (I + BK)\Sigma_{op}(I + BK)^T E_N^T) \Sigma_{\text{lim}} \geq 0.
\]
(37)
The second term of inequality (37), that can be denoted by \(\Xi\), is symmetric and, hence, diagonalizable via orthonormal matrix \(S\), making the previous formulation become \(S(I - \text{diag}(\lambda_1(\Xi), \ldots, \lambda_n(\Xi)))S^T \geq 0\), solved by \(\lambda_{\text{max}}(\Xi) \leq 1\). Similarly to what has been done in the chance constraint, we can say that \(\lambda_{\text{max}}(\Xi) \leq ||M(I + BK)^T E_N^T \Sigma_{\text{lim}}^{-1/2}||^2\), that brings to the final form of the constraint
\[
||M(I + BK)^T E_N \Sigma_{\text{lim}}^{-1/2}||^2 \leq 1.
\]
(38)
3) State constraint: The acceleration of the engine is limited by both an upper and a lower limit, i.e., \(a_{\text{min}} \leq a \leq a_{\text{max}}\) for \(i = 1, \ldots, N\). This requirement is added to the cost function as a soft constraint through the addition of a logarithmic barrier function that makes the cost function go to infinity as the state approaches the limit. This function is expressed as
\[
S_a = \sum_{k=1}^{N} (-\log(a_{\text{max}} - a_k) - \log(-a_{\text{min}} + a_k)),
\]
(39)
where \(a_k\) is a function of \(K\) and is given by
\[
a_k = ((I + BK)(A\hat{x}_0 + D\xi + Ee_0))^T \begin{bmatrix} 0 \in \mathbb{R}^{1 \times (kn_x - 1)} \\ 0 \in \mathbb{R}^{1 \times (N-k)n_x} \end{bmatrix}.
\]
(40)
\(^{1}\)Only for \(p_f < 0.5\) so that \(\Phi^{-1}(p_f) < 0\), but the other cases don’t have a reason to be explored since real-life applications are for much lower failure probabilities.
F. Problem formulation

The final problem formulation is obtained by minimizing the cost function of $K$ by using the obtained constraints:

$$
\min_K J(K) + (39)
\text{s.t. } (36), (38).
$$

(41)

It can be easily shown that the logarithmic barrier functions are convex (by, e.g., second derivative) with respect to $K$. Thus, adding them to the cost function doesn’t affect its convexity.

G. Algorithm for the shrinking horizon control

For this controller to be robust against noise realisation that can change the conditions from the expected ones, a MPC-style cyclic algorithm is implemented, having the problem solved at every time step and the time horizon $N$ that shrinks, as shown in Algorithm 1.

\begin{verbatim}
Algorithm 1: Robust controller for intersection crossing

1: Input: $\dot{x}_0$, $\Sigma_0$, $p_d$, $p_f$, $A, B, C, D, G$, $s_{exit}$, $lim$, $\Sigma_{lim}$, horizon length $N$.
2: Initialization: failures counter $\bar{f} = 0$, observation ensemble $K$, $\Sigma_0 = \Sigma_0$,
   since the last successful observation $\delta = 0$.
1: while $N > 0$ do
2:   Generate $\gamma \in \mathbb{R}^N, 0 < \gamma_i < 1$ for $i = 1, ..., N$. Solve (41) for
   $\gamma$ obtained.
3:   if measurement $\tilde{y}_1$ arrives at step $k = 1$ then
4:     $\delta_{1} = 1$
5:   else if measurement $\tilde{y}_1$ doesn’t arrive at $k = 1$ then
6:     $\delta_{1} = 0$
7: end if
8:   if there is a feasible solution for (41) then
9:     Save $\bar{u} \leftarrow u$, $\bar{K} = K$, $\Sigma_0 \leftarrow \Sigma_0$,
   the last combination that provided a successful optimization.
10: Save $\delta \leftarrow \delta_{1}$, $\bar{f} \leftarrow 0$.
11: Compute $\Sigma_1$ using (6a)-(6c) with $\Sigma_{k|k} = \Sigma_0$ and
   $K_{k|k} = K_{\delta_{i}}$, $i = 1, ..., n_u$.
12: Compute $\Sigma_1$ using (6a)-(6c) with $\Sigma_{k|k} = \Sigma_0$ and
   $K_{k|k} = K_{\delta_{i}}$.
13: $\Sigma_{k+1} = (\delta F_i C A )\Sigma_0 (\delta F_i C A)^T + \Sigma_0 \delta_{1}$
   (obtainable from the observation evolution).
14: $\hat{x}_1 = N(A \hat{x}_1 + B \bar{u}_1)$, $\delta_{1} = \Sigma_{k1}$.
15: else if the optimization fails then
16:     $\delta \leftarrow \delta + \delta_{1}$, $\bar{f} \leftarrow \bar{f} + 1$.
17: Compute $\Sigma_1$ using (6a)-(6c) with $\Sigma_{k|k} = \Sigma_0$ and
   $K_{k|k} = K_{\delta_{i}}$
18: Compute $\Sigma_{k1}$ as $\Sigma_{k1}(\bar{f} + 1)$ using $\Sigma_0$, $\bar{K}$, and $\delta$.
19: $\hat{x}_1 = N(A \hat{x}_1 + B \bar{u}_{\bar{f} + 1})$, $\delta_{1} = \Sigma_{k1}$.
20: end if
21: $\dot{x}_0 \leftarrow \hat{x}_1$, $\Sigma_0 \leftarrow \Sigma_1$.
22: end while
Output: Control sequence that solves the problem at every iteration.
\end{verbatim}

Notice that the procedure of having a failures counter and saving the observation history from the last successful implementation is necessary due to the matrix $K$ structure: in fact, the covariance introduced by the control action is a pure feedback only for the first control step. After it, it starts taking into account also the previous steps. Thus, to have the right covariance evolution, it has to be constructed starting from the last successful step, since that same control sequence is used until communication succeeds.

IV. Numerical Examples

A. Simulation Parameters

The simulations have been run with both sampled and direct distributions and making both the packet drop probability and the accepted failure probability variate to investigate their effect on the solution. The tool used to solve the constrained optimization problem is the CVX MATLAB toolbox. The simulations have been run for a total time of 4 seconds, divided in $N=20$ intervals of $h = 0.2s$ each, and the acceleration-to-deceleration factor is set to $\tau = 10$. The process noise matrix is set to $D = \text{diag} [0.5, 0.05, 0.05]$, while the measurement noise matrix is $G = 0.5D$. The Initial error covariance is initialized to $\Sigma_0 = \text{diag} [1, 0.1, 0.1]$, and the final limit for the estimation covariance is $\Sigma_{lim} = 0.5\Sigma_0$. The cost matrices are respectively $Q = \text{diag} [1, 0.01, 5]$ for the state deviation and $R = 5$ for the input.

B. Sampled Probability Distribution

1) Effect of the failure probability allowed: The results for $k = 0$ (at the first time instant) are shown. The effect of a decrease of the allowed failure probability $p_f$ while keeping $p_d = 0.5$ are shown in Figure 1. In order to have comparable results the $\delta$ ensemble for the times instant for $k = 1,...,20$ is not a random one but it’s intermittent (i.e. $\delta = [1 0 1 0 1 ...]$). It can be seen that the expected value of the final position increases for decreasing $p_f$ in order to make the probability distribution of the final position match the increasingly demanding requirement of the chance constraint. For $p_f = 0.49$ the expected value of the position is almost equal to the exit section, while as $p_f$ decreases, either the expected value is increased or the covariance is shrunk to a smaller value.

2) Effect of the packet drop probability: The effect of an increase of $p_d$ while keeping $p_f = 0.0005$ for the first time instant is shown in Figure 2. The control sequence becomes more conservative as $p_d$ increases, since the covariance (especially $\Sigma_{k}$ increases. Also in this case there is an influence given by the $\delta$ sequence used for the simulation: in fact, since once an error enters the estimation following a successful communication ($\delta = 1$), then that error freely evolves in the following iterations according to (8), and so does its covariance. Consequently, if a high-covariance error is obtained at a certain step and in the following iteration there is a successful communication, the effect on the final $\Sigma_k$ makes difficult to steer it. This may even bring to unfeasible problems for high $p_d$ when a sample is obtained with all the successful communications in the last time steps. In the following section, the effect of the communication sequence will be further investigated.

C. Sampled vs. Direct Probability Distribution

In this section, a comparison is made between the more realistic sampled distribution of the communication ensemble and the direct probability distribution in order to evaluate if the direct probability distribution case is a reliable alternative. The input sequence obtained at the initial time step for the direct distribution is compared to the one obtained with a random sampled distribution with the same $p_f$ for different values of $p_d$, and the results are shown in Figure 3. While the control sequence with the direct probability distribution stays very similar for increasing $p_d$ (as discussed in the previous section, the notable increase of the cost function value is obtained for $p_d$ higher than 0.8), the sampled probability distribution control sequence visibly becomes more conservative. The conclusion is that the decision horizon $N$ is too small to realistically use a direct probability distribution to represent the future ensemble.
(a) Position (expected value and 1σ interval).  
(b) Acceleration  
(c) Input power
Fig. 1: First iteration of the optimization problem with sampled probability packet drop distribution at $p_d=0.5$ and variable $p_f$.

(a) Position (expected value and 1σ interval).  
(b) Acceleration  
(c) Input power
Fig. 2: First iteration of the optimization problem with sampled probability packet drop distribution at $p_f=0.0005$ and variable $p_d$.

(a) $p_d=0.2$  
(b) $p_d=0.4$  
(c) $p_d=0.8$
Fig. 3: First iteration of the optimization problem with direct probability packet drop distribution at $p_f=0.0005$ and variable $p_d$.

D. Application to the Shrinking Horizon

In this section, Algorithm 1 is applied to the problem with the initial condition described before, $p_f=0.0005$ and variable $p_d$. The results obtained are visible in Figure 4. Even though the simulations shown here have a successful outcome (i.e. a successful last step), there are some cases where it doesn’t happen. Some of the possible causes are: 1) The first communication does not have the same outcome obtained in the randomly generated ensemble, this causes the initial covariance of next iterate to be different from the one expected in the previous one. The probability for this to happen are equal to $p_d(1-p_d)$, so it is a phenomenon that can’t happen for $p_d=\{0,1\}$ and is maximized at $p_d=0.5$. This theory is reinforced by the fact that for $p_d$ near to 0.5 the failure ratio is higher and the reason why in [22] an observer with delay is treated. 2) The randomly generated ensemble makes the estimation covariance impossible to steer. 3) The noise generates initial conditions that are unsolvable while staying within the constraints. This can be avoided by constraining all the states but the initial one, so that the controller can steer the initial state in the feasibility region.

It is possible to see that the more $p_d$ increases, the more is likely to have two subsequent control sequences that are the exact same. On the other hand, for high $p_d$ when a sequence changes it tends to detach from the previous one in a more evident way.

E. Performance statistics

In order to assess the statistic performance of the algorithm, 30 simulations have been made for each different $p_d$. The steps where the optimization failed at every simulation have been collected as well as the simulation which last iteration had a successful result (The trivial case of $p_d=1$ is here neglected). The results are shown in Table I. It can be seen that the percentage of success decreases for increasing $p_d$. This happens slightly for $p_d$ up to 0.8, while for $p_d=0.9$ the decrease is more evident.

V. CONCLUSIONS AND FUTURE DIRECTIONS

In this work, the problem of intersection crossing for autonomous vehicles passing through an intersection has been considered. The control input computation is outsourced to a central coordinator that receives the measurement from the vehicle through an unreliable communication channel with a given information drop probability. A theoretical control algorithm has been formulated solving at every
The coordination of multiple vehicles simultaneously is a step towards making this kind of algorithms applicable, as well as the application of it to a steering car which is more challenging, since the dynamic system in that case is not time-invariant (unless some simplifications are made).

### REFERENCES


