Annual Reviews in Control 52 (2021) 543–553

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Full Length Article

Convex output feedback model predictive control for mitigation of COVID-19 pandemic[☆]

T. Péni [a,*], G. Szederkényi [b]

[a] *Institute for Computer Science and Control (SZTAKI), Eötvös Lóránd Research Network (ELKH), H-1111, Kende u. 13-17., Budapest, Hungary*
[b] *Pázmány Péter Catholic University, H-1083 Práter u. 50/a, Budapest, Hungary*

ARTICLE INFO

Keywords:
Model predictive control
State estimation
Geometric programming
Compartmental models

ABSTRACT

In this paper, a model predictive control approach is proposed for epidemic mitigation. The disease spreading dynamics is described by an 8-compartment smooth nonlinear model of the COVID-19 pandemic in Hungary known from the literature, where the manipulable control input is the stringency of the introduced non-pharmaceutical measures. It is assumed that only the number of hospitalized people is measured on-line, and the other state variables are computed using a state observer which is based on the dynamic inversion of a linear sub-system of the model. The objective function contains a measure of the direct harmful consequences of the restrictions, and the constraints refer to input bounds and to the capacity of the healthcare system. By exploiting the special properties of the model, the nonlinear optimization problem required by the control design is reformulated to convex tasks, allowing a computationally efficient solution. Two approaches are proposed: the first finds a suboptimal solution by geometric programming, while the second one further simplifies the problem and transforms it to a linear programming task. Simulations show that both suboptimal solutions fulfill the design specifications even in the presence of parameter uncertainties.

1. Introduction

The COVID-19 pandemic is one of the biggest challenges the world is currently facing with. Until the vaccine and effective treatment are widely available, carefully planned measures have to be introduced in every country to avoid the serious effects of the spreading disease. Choosing a right management policy is a sensitive task where several potentially contradicting objectives have to be taken into consideration. The most important limiting constraint is the capacity of the healthcare system, which can be easily overwhelmed if the spread of the disease is not controlled. It is clear that the transmission of the virus can be efficiently slowed down by appropriate restrictions (social distancing, lockdown), but these measures have negative impacts on the society and the economy that cannot be neglected. At the moment, governments are continuously evaluating the control measures, trying to find balance between public health concerns and the costs of social distancing measures.

The population-level dynamical modeling of disease spread is an intensively studied area with a large number of applicable model structures corresponding to different assumptions Brauer, Castillo-Chavez, and Castillo-Chavez (2012). There is also a wide literature on epidemic control design where the input can comprise vaccination and/or non-pharmaceutical measures Anderson and May (1992), Nowzari, Preciado, and Pappas (2016). Among the possible techniques, model predictive control (MPC) is a straightforward approach for handling the often contradicting goals and strict constraints in this context, therefore, the systematic design of non-pharmaceutical interventions against COVID-19 using MPC has been addressed in several recent publications. In Köhler et al. (2020) it was shown that the number of fatalities can be efficiently reduced using an adaptive feedback strategy to compute the stringency of social distancing even if there are significant parametric and input uncertainties in the model. At the same time, it is also clearly illustrated in the paper that neglecting the feedback

[☆] G. Szederkényi acknowledges the partial support of the Thematic Excellence Programme (TKP2020-NKA-11). This work was partially supported by the National Research, Development, and Innovation Office, Hungary through the grant NKFIH-OTKA-131545. Research related to T. Péni was supported by the Ministry of Innovation and Technology NRDI Office, Hungary within the framework of the Autonomous Systems National Laboratory Program.

The authors thank Balázs Csutak and Gergely Horváth for their help in the model and control loop analysis. The authors are grateful to the anonymous reviewers for their constructive comments.

[*] Corresponding author at: Institute for Computer Science and Control (SZTAKI), Eötvös Lóránd Research Network (ELKH), H-1111, Kende u. 13-17., Budapest, Hungary.

E-mail addresses: peni@sztaki.hu (T. Péni), szederkenyi@itk.ppke.hu (G. Szederkényi).

https://doi.org/10.1016/j.arcontrol.2021.10.003
Received 29 March 2021; Received in revised form 7 July 2021; Accepted 5 October 2021
Available online 27 October 2021

principle might be dangerous in handling the pandemic. In Morato, Bastos, Cajueiro, and Normey-Rico (2020) a binary input representing full or no isolation is computed using MPC and a nonlinear epidemic model to reduce the peak of infected individuals while minimizing the economic loss due to the restrictions. This solution is refined and made more realistic in Morato, Pataro, da Costa, and Normey-Rico (2020), where weekly piecewise constant input with actuation dynamics are assumed. In Péni, Csutak, Szederkényi, and Röst (2020), a nonlinear MPC-based solution was proposed for epidemic management, where the main contribution is the application of temporal logic for the incorporation of interdependent and possibly time-varying constraints into the design.

As the plenty of successful solutions show, nonlinear MPC is a suitable tool to transform the various control goals and objectives related to the epidemic management into a transparent optimization problem. However, the numerical solution of this problem requires significant computational power even in the case of a simple 4–8 state compartment model. This makes it difficult to extend the algorithms for large dimensional (e.g. multi-scale or multi-region) epidemic models, to compute control sequence for long horizons and to apply advanced robust control design algorithms (e.g. tube or scenario-based methods). Therefore, an important aim of this work is to reformulate the MPC design in such a framework, where the numerical computations can be performed more efficiently. From this respect, a recent related work is Hayhoe, Barreras, and Preciado (2020), where a data-driven model is proposed to describe the spread of the COVID-19 pandemic, and an optimal controller is designed in a convex framework using geometric programming to minimize the final number of deaths while taking into consideration cost and hospital capacity constraints. Moreover, a detailed and realistic actuation model identified from mobility data is also presented in Hayhoe et al. (2020).

A well-known characteristic of predictive control is that it generates a full state feedback policy, which can be applied only if all of the states are measured or estimated. In the work of Péni et al. (2020) the possibility of output feedback control was demonstrated by designing a state observer for the epidemic model. In this paper an improved state estimator is proposed, which by exploiting the specific structure of the compartmental model can be constructed in an elegant way and provides reliable state estimates even in the presence of modeling uncertainty. The observer and the convexified nonlinear model predictive controller give an output feedback structure that is proposed as a possible solution to the control problem related to the mitigation of the pandemic.

2. Compartmental model

The transmission dynamics of the epidemic is usually described by compartmental models where each state variable represents one group of the population. The disjoint groups collect individuals of the same infectious status. Depending on the modeling assumptions and goals, different compartmental models can be constructed Ansumali, Kaushal, Kumar, Prakash, and Vidyasagar (2020), Carli, Cavone, Epicoco, Scarabaggio, and Dotoli (2020). In this paper we use the 8 state model introduced in Péni et al. (2020). This specific model structure has been developed to capture the dynamic characteristics of the epidemic that are relevant to our control design objectives. In the model construction it has been important that all parameters have clear physical interpretation and they can be determined with sufficient precision by using the available medical literature and epidemiological data.

In our model the total population is divided into 8 classes: S denotes the susceptibles, i.e. those who can be infected by the disease, L (latent) collects individuals, who have already contracted the disease but do not show symptoms and are not infectious yet; P is the pre-symptomatic compartment for those who are infected, but they need a few days to develop any symptoms. After the incubation period elapses the infected individuals are transferred to the asymptomatic (A) and symptomatic (I) compartments. Those in A will always recover, while the more severe cases in I may require hospitalization and move to compartment H. Patients in H may eventually recover (R) or die (D). The transition diagram is depicted in Fig. 1, further details on the model can be found in Péni et al. (2020).

Table 1
Model parameters and their nominal values.

Parameter	Interpretation	Nominal value
α^{-1}	Latent period	2.5 (days)
p^{-1}	Pre-symptomatic infectious period	3 (days)
β	Transmission rate	1/3
δ	Relative transmissibility of asymptomatic	0.75
q	Prob. of developing symptoms	0.6
ρ_I^{-1}, ρ_A^{-1}	Infectious period	4 (days)
η	Hospitalization probability of symptomatic cases	0.076
h^{-1}	Average length of hospitalization	10 (days)
μ	Probability of fatal outcome, given hospitalization	0.145

The differential equations of the compartmental model are written as

$$\dot{x}_S = -\beta\, v \left[x_P + x_I + \delta x_A\right] x_S, \tag{1a}$$

$$\dot{x}_L = \beta\, v \left[x_P + x_I + \delta x_A\right] x_S - \alpha x_L, \tag{1b}$$

$$\dot{x}_P = \alpha x_L - p x_P, \tag{1c}$$

$$\dot{x}_I = q p x_P - \rho_I x_I, \tag{1d}$$

$$\dot{x}_A = (1-q) p x_P - \rho_A x_A, \tag{1e}$$

$$\dot{x}_H = \rho_I \eta x_I - h x_H, \tag{1f}$$

$$\dot{x}_R = \rho_I (1-\eta) x_I + \rho_A x_A + (1-\mu) h x_H, \tag{1g}$$

$$\dot{x}_D = \mu h x_H. \tag{1h}$$

The state variables represent the number of individuals in the compartments and are assumed to be normalized to 1, i.e. $\sum_{i\in\{S,L,P,I,A,H,R,D\}} x_i = 1$, where 1 represents the total population. By construction, $\sum_{i\in\{S,L,P,I,A,H,R,D\}} \dot{x}_i = 0$ so the size of the population remains constant. The parameters of the model are summarized in Table 1. The nominal values are based on the estimates calculated from the epidemiological data collected in Hungary and also worldwide. For more details, see Péni et al. (2020), Röst et al. (2020).

Until vaccine or appropriate treatment is available, the spread of the virus can only be controlled by limiting the contacts and reducing the infection probability between individuals. The effect of different interventions (from mandatory mask wearing to total lockdown) can be quantified by different scalings of the transmission rate β, therefore it is straightforward to choose the scaling factor v as control input. Assuming that the number of hospitalized and deceased individuals can be reliably documented, state variables x_H and x_D are chosen for measured state variables. The official number of daily new infections and active cases cannot be used directly, since it is known that only a fraction of the actual cases are detected Phipps, Grafton, and Kompas (2020).

The observability of the model was briefly analyzed in an LPV framework in Péni et al. (2020) where it was found that the model is observable from x_H, although the observability matrices are numerically badly conditioned. Following the methodology of Massonis, Banga, and Villaverde (2020) and using the STRIKE-GOLDD software tool Villaverde, Barreiro, and Papachristodoulou (2016), the identifiability of model parameters was also checked with the assumption that x_H and x_D are observed. Computation results showed that the model itself is not structurally identifiable, since the parameters δ and q cannot be uniquely determined by measuring only x_H and x_D. This is an important fact from the point of view of model analysis and parameter estimation, but does not affect the solvability of the control task.

Discretization. Adapting to the predictive control framework, the epidemic model will be used later in discrete time. To discretize the

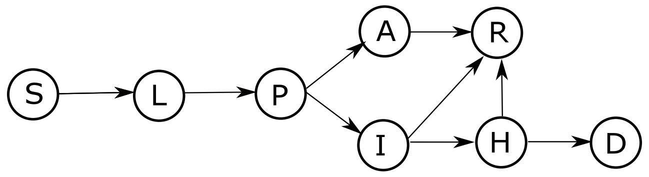

Fig. 1. Transition diagram. Circles represent compartments and arrows represent transitions between the compartments.

dynamics, we make the following observation: (1) can be considered as a linear time invariant (LTI) system driven by a nonlinear input function:

$$\dot{x} = Ax + B \cdot \upsilon(x_P + x_I + \delta x_A)x_S \tag{2}$$

$$y = Cx \tag{3}$$

where

$$B = \begin{bmatrix} -\beta & \beta & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

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

and A can be determined from (1). Vector x comprises the states. A discrete time model can be obtained by applying standard zero order hold (ZOH) method to the LTI part and evaluating the input function only at the discrete time steps. The discretized model can be given formally as follows:

$$x^{(k+1)} = A_d x^{(k)} + B_d \cdot \upsilon^{(k)}(x_P^{(k)} + x_I^{(k)} + \delta x_A^{(k)})x_S^{(k)} \tag{4}$$

We will use two different sampling times in our study. To obtain acceptable predictions for the considered long horizon, the predictive controller (Section 4.) uses $T_s = 0.5$ day for updating the states, while $T_s = 1$ day is used for the observer design (Section 3). We further assume that the update of the input can be performed weekly.

3. State reconstruction

The goal of the state observer is to reconstruct the states of model (1) from the available output measurements. The proposed observer is designed in 4 steps, these steps are presented in detail in this section. The observer is designed in discrete time and as we have mentioned above, the sampling time is $T_s = 1$ day. This means that the output measurements can be updated only on a daily basis, which is feasible in practice. It is important to emphasize that the observer is designed for the nominal model, possible parameter uncertainites are not considered explicitly in the design. On the other hand, in Section 5, the observer is tested on uncertain models to analyze its properties on realistic scenarios.

3.1. Inversion based estimator for x_L

We start from the LTI model defined by Eqs. (1c)–(1f), where state x_L can be considered as an unmeasured external input and x_H is the measured output. Let this dynamical system be converted to discrete time domain and let the resulting transfer function be denoted by $G_{LH}(z)$. One possible approach to reconstruct x_L is to invert this dynamics and drive the inverse with input x_H. If the model is accurate enough and a causal, stable inverse exists, then a reliable estimate $\hat{x}_L$ can be obtained. Note also that the other 3 state variables are not needed for inversion because the system is linear, so the inverse can be started from any (e.g. zero) initial state.

By examining $G_{LH}(z)$ it turns out that the system is strictly proper and contains unstable zeros. Therefore, the construction of the inverse requires the numerical differentiation of the output and the inverse obtained will not be stable. Fortunately, the first problem is easy to handle. By computing the Hankel singular values of the $x_L \rightarrow x_H$ dynamics we obtain the following result: [0.1175, 0.0289, 0.0026, 0]. A zero singular value is present because x_A does not affect the $x_L \rightarrow x_H$ transfer. This state can therefore be trivially removed from the model. The third singular value is significantly smaller than the other two, so we expect that the removal of the corresponding state does not generate significant error. Doing so, two states are removed from the model. By performing a balanced reduction (function 'balred' in MATLAB) we obtain the following 2-state, proper transfer function:

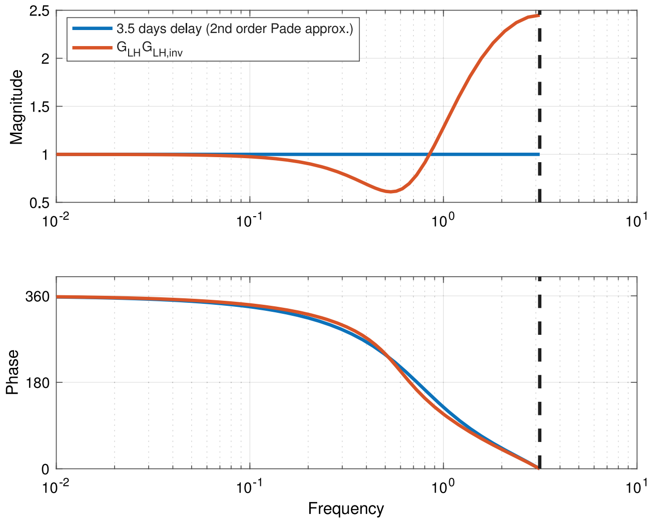

Fig. 2. Comparison of $G_{LH}(z)G_{LH,red,inv}(z)$ with a 2nd order Pade approximation of 3.5 days time delay.

$$G^r_{LH}(z) = \mathrm{tf}(z_1, z_2, p_1, p_2) := \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \tag{5}$$

where $p_1 = 0.92787$, $p_2 = 0.94446$ and $z_{1,2} = 1.27205 \pm 0.37353i$. (The numerical values of the poles and zeros have been obtained by using the nominal model parameters in Table 1.) Note that $G^r_{LH}(z)$ can be easily inverted, without numerical differentiation.

The second problem (instability of the inverse) still holds, because $G^r_{LH}(z)$ has a complex, unstable zero pair. To make the inverse stable, the zero is transformed back to continuous time by $\tilde{z}_{1,2} = \log(z_{1,2})/T_s$, then reflected across the imaginary axis and finally it is transformed back to discrete time by $\bar{z}_{1,2} = \exp(\tilde{z}_{1,2}T_s)$. Then $G^{r,i}_{LH}(z) = \mathrm{tf}(p_1, p_2, \bar{z}_1, \bar{z}_2)$ is considered as the stable, approximate inverse of $G^r_{LH}(z)$. Due to the reduction and the transformation of the zeros, $G^{r,i}_{LH}(z)$ does not properly cancel $G_{LH}(z)$. The product of them generates a phase shift corresponding to $t_d = 3.5$ days time delay. This can be seen on the bode diagram in Fig. 2, where $G_{LH}(z)G^{r,i}_{LH}(z)$ is compared with the 2nd order Pade approximation of the $t_d = 3.5$ days time delay. It can also be seen that the magnitude of $G_{LH}(z)G^{r,i}_{LH}(z)$ significantly differs from 1 at higher frequencies. Since the epidemic model is controlled by piecewise constant input that changes relatively rarely compared to the sampling time, this approximate inverse will be suitable for computing an estimate for x_L.

3.2. Reconstruction of x_P, x_I, x_A

By using $G^{r,i}_{LH}(z)$, we can obtain at time $t = k \cdot T_s$, the delayed estimate $\hat{x}_L(t - t_d)$. Using this input, together with the measured, delayed output $x_H(t - t_d)$ a state observer can be designed for the state–space counterpart of $G_{LH}(z)$. In this paper a standard Kalman filter is proposed for this purpose. The Kalman filter also has the practical advantage of being able to handle state and measurement noise, which are typically present in a real system. By using the time shifted inputs, the Kalman filter can produce estimate for the delayed states, that is, we obtain $\hat{x}_P(t - t_d)$, $\hat{x}_I(t - t_d)$ and $\hat{x}_A(t - t_d)$.

3.3. Estimation of x_S and computing the actual state values

In this step the first 5 equations of the epidemic model are numerically integrated on time interval $[t - t_d, t]$. The initial state is $[\hat{x}_S^-,\ \hat{x}_L(t-t_d),\ \hat{x}_P(t-t_d),\ \hat{x}_I(t-t_d),\ \hat{x}_A(t-t_d)]$, where $\hat{x}_S^-$ represents $\hat{x}_S(t-t_d)$ and is regularly updated as we describe later. At the beginning it is initialized to 1 (assuming that the state estimation starts at the beginning of the epidemic). The integration of the model requires also the past control input sequence that has been applied on time interval $[t - t_d, t]$. This control sequence is assumed to have been previously stored and is available.

The integration of the model is performed in two steps. First the state trajectories are computed on interval $[t-t_d,\ t-t_d+T_s]$ to obtain $\hat{x}_S(t-t_d+T_s)$, which is used to update $\hat{x}_S^-$. Continuing the integration on time interval $[t-t_d+T_s,\ t]$ the actual values of the state estimates, i.e. $\hat{x}_S(t)$, $\hat{x}_L(t)$ $\hat{x}_P(t)$, $\hat{x}_I(t)$ and $\hat{x}_A(t)$ can be obtained. The compartment of recovered patients will not be used later in the control design, but if it is necessary to estimate x_R, it can be easily done by using the estimated states and measured output x_D: as the sum of the states is 1, we have $\hat{x}_R(t) = 1 - \sum_{i\in\{S,L,P,I,A\}} \hat{x}_i(t) - x_D(t) - x_H(t)$.

The complete algorithm of state reconstruction is summarized in Algorithm 1.

Algorithm 1 State estimation algorithm

```
$t := 0$, $T_s = 1$, $\hat{x}_S^- = 1$
repeat
    1. Measure $x_H(t)$
    2. Compute $\hat{x}_L(t-t_d)$ by updating the states and the output of $G_{LH}^{r,i}$ with input $\hat{x}_H(t)$.
    3. Compute $\hat{x}_P(t-t_d)$, $\hat{x}_I(t-t_d)$ and $\hat{x}_I(t-t_d)$ by updating the states of the Kalman filter with input $\hat{x}_L(t-t_d)$ and output $x_H(t-t_d)$.
    4. Integrate dynamic equations (1a)-(1f) on time interval $[t-t_d,\ t-t_d+T_s]$ by using the past control input $v(\tau)$, $\tau \in [t-t_d,\ t-t_d+T_s]$. Set $\hat{x}_S^- = \hat{x}_S(t-t_d+T_s)$.
    5. Continue the integration of the model on time interval $[t-t_d+T_s,\ t]$ to obtain state estimation $[\hat{x}_S(t), \hat{x}_L(t), \hat{x}_P(t), \hat{x}_I(t), \hat{x}_A(t)]$.
    6. Compute $\hat{x}_R(t)$ by using measurement $x_D(t)$
    7. $t := t + T_s$
until needed
```

We remark that another possible approach is to design an unknown input observer to estimate x_P, x_I, x_A and x_L using the theory in Bhattacharyya (1978). However, for this the first and some higher derivatives of x_H are also needed as auxiliary outputs to fulfill the computability conditions.

4. Nonlinear predictive control by convex optimization

4.1. Control problem formulation

The epidemic model has one manipulable input: the scalar factor v that scales the transmission rate β (see, Eq. (1)). This variable represents the effect of the measures that are applied on the society in order to control the contact rate between individuals. Regarding the control goals, several scenarios can be reasonable. We are focusing on the mitigation of the epidemic, where the goal is to avoid the overwhelming of the healthcare system in a finite time window by applying less stringent interventions whenever possible. Strict measures have serious adversarial effects on society and economy so avoiding them is desirable Mandel and Veetil (2020), Saladino, Algeri, and Auriemma (2020). The time window is typically chosen to be long enough to make acceptable long-term prediction for the future behavior of the epidemic. It is also an option to align it to the estimated date when a suitable vaccine or treatment becomes available.

Following a similar concept as in our earlier paper Péni et al. (2020), we formulate a nonlinear model predictive controller for the mitigation problem above. The control input is computed periodically based on the most recent measurements and the predicted future behavior of the model. Since the control window is fixed (not shifted in each time step), a shrinking horizon MPC is designed. The controller is implemented in discrete domain by using $T_s = 0.5$ sampling time. At a time instant $t = k \cdot T_s$ the control input is computed by solving the following optimization problem:

$$\max_{v_0 \dots v_{M(k)-1},} \sum_{i=0}^{M(k)-1} w_i v_i \tag{6a}$$

$$\text{w.r.t.} \tag{6b}$$

$$x^{(i+1)} = A_d x^{(i)} + B_d \cdot v^{(j)} (x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)}) x_S^{(i)} \tag{6c}$$

$$x^0 = x_k \quad \text{(state at time } k\text{)}, \tag{6d}$$

$$x_H^{(i)} \le \bar{x}_H \tag{6e}$$

$$i = 0 \dots N(k) + K - 1, \tag{6f}$$

$$j = \begin{cases} \lfloor i/L \rfloor & \text{if } i \le N(k) \\ \lfloor N(k)/L \rfloor & \text{if } i > N(k) \end{cases} \tag{6g}$$

$$\underline{v} \le v_\ell \le 1, \quad \ell = 0 \dots M(k) - 1 \tag{6h}$$

where $x^{(i)} = x^{(k|k+i)}$, $v^{(i)} = v^{(k|k+i)}$ are the predicted state and control input at the $k+i$th time step and $x^{(0)} = x^{(k|k)} = x(kT_s) = x(t)$ is the state at the current time instant. Weights w_i (with $0 \le w_i$) are used to differently penalize control actions at different time instants. They are a-priori fixed tuning parameters.

Strict interventions are penalized by maximizing the cost function (6a). To protect the functionality of the healthcare system a strict upper bound is introduced for the number of hospitalized patients (6e). Further constraints are added to keep the control input in a valid range (6h). The maximal value of v is 1 (no intervention), the minimal value $\underline{v}$ is chosen to correspond to the strictest measure, e.g. the total lockdown.

Since the control input determines rules and restrictions that have to be performed by the society, it takes time to have impact on the dynamics. Therefore it is no use changing the control input at each sampling instant. We therefore assume that v is updated only at every Lth time step (6g).

Assume the control window is $[0, H]$, where H is a priori fixed. The prediction horizon is $N(k) + K$, where $N(k) = H - k$ is shrinking as k moves towards H. K defines a short time interval after the control window, where the control input is not updated, but the constraints have to be satisfied. This prevents the optimization to simply improve the cost function by generating less effective (in our case higher) control actions at the end of the horizon. Turning off the controller in the last few steps leads to high increase in the constrained state variable, which can therefore easily exceed the limit right after leaving the control window. This safety constraint can be used together with a small weight on the last control action (e.g. $w_{M(k)} < \min_{i \ne M(k)} w_i$) allowing strict intervention at the end of the control window. This procedure implements the following concept: the system is left at time H in a state, where there exist (at least a strict) intervention that is able to keep the states below the prescribed limits at least for $K \cdot T_s$ time period after the end of the control window.

The last parameter of the procedure is $M(k) = \lfloor N(k)/L \rfloor + 1$, which denotes the number of free control moves to be designed.

Remark 1. Since compartments of recovered (R) and deceased (D) patients are not used in the control synthesis process, the corresponding state variables x_R and x_D can be removed from the control model.

4.2. Suboptimal solution by geometric programming

The optimization problem (6) is nonlinear and nonconvex. In addition, the control horizon is long, considering that the control window is at least about six months long, but might be even longer. These factors make the control design computationally challenging. Our goal is therefore to recast the original problem to a convex optimization task. For this, the geometric programming (GP) framework Boyd, Kim, Vandenberghe, and Hassibi (2007), Boyd and Vandenberghe (2009) will be used. Geometric programming is a powerful tool to solve nonlinear and nonconvex optimization problems if the decision variables are positive and the cost and the constraints are polynomial functions with positive coefficients. More precisely, a geometric program in standard form can be given as follows:

$$\min \ f_0(x) \tag{7a}$$

$$\text{s.t.} \ f_i(x) \le 1 \tag{7b}$$

$$g_i(x) \le 1 \tag{7c}$$

where $x \in \mathbb{R}^n$ with $x_j > 0$, $\forall j \in \{1,\dots,n\}$; $g_i(x)$-s are monomials, i.e. $g_i(x) = c_i x_1^{a_{i,1}} x_2^{a_{i,2}} \dots x_n^{a_{i,n}}$ with $c_i > 0$ and $a_{i,j} \in \mathbb{R}$; $f_i(x)$-s are posynomials (sums of monomials), i.e. $f_i(x) = \sum_{k=1}^{K_i} c_k x_1^{a_{k,1}} x_2^{a_{k,2}} \dots x_n^{a_{k,n}}$, $c_k > 0$ and $a_{k,j} \in \mathbb{R}$. The main advantage of GP is that (7) can be systematically transformed to a nonlinear convex optimization problem by taking the logarithm of the variables. It is clear that this transformation converts (7c) to linear constraints, but it can also be shown that the logarithm of the posynomials are also convex in the transformed variables. The standard GP can be extended by allowing further functions of posynomials, e.g. max, positive (fractional) power, etc. in the construction of the optimization problem. It can be shown that these operators preserve the convexity. The convex counterpart of a standard (or extended) GP can be constructed manually or by using a suitable software tool, e.g. the CVX toolbox Grant and Boyd (2020). More details on GP and its applications can be found in the paper and book Boyd et al. (2007), Boyd and Vandenberghe (2009).

In the case of epidemic control, the GP method is motivated by the following properties of (6):

- the model is polynomial in the state and input
- all state variables are inherently positive (nonnegative),
- apart from the dynamics of x_S, the coefficients of every term in the discrete time model are nonnegative.

To eliminate the problem caused by the dynamics of x_S, we can make the following observations: first, x_S continuously decreasing; second, a successful mitigation control keeps the number of infections low, so only a limited portion of the society is expected to get infected by the end of the control window. Therefore, $x_S^{(i)}$ in (6) can be replaced by its constant upper bound $\bar{x}_S = x_S^{(0)}$. By using this upper bound, stricter control actions are generated that may degrade the performance (results in larger control cost). However, the simulations will show that this is an acceptable price to get a significantly simpler optimization task.

Note also that the original cost function cannot be transformed in the geometric programming framework. Therefore, it is replaced by $\sum w_i(1/v_i)$. Of course, this new cost is not equivalent to the original one, but expresses the same intention: minimizes the strictness of the interventions. The modified optimization problem can be given as follows:

$$\min_{v_0 \dots v_{M(k)-1},} \quad w_\varepsilon \varepsilon + \sum_{i=0}^{M(k)-1} w_i \frac{1}{v_i} \tag{8a}$$

$$\text{w.r.t.} \tag{8b}$$

$$\tilde{x}^{(i+1)} = \tilde{A}_d \tilde{x}^{(i)} + \tilde{B}_d \cdot v^{(j)}(x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)})\bar{x}_S \tag{8c}$$

$$x^0 = x_k \quad \text{(state at time } k\text{)}, \tag{8d}$$

$$x_H^{(i)} \le \bar{x}_H \varepsilon \tag{8e}$$

$$\varepsilon \ge 1 \tag{8f}$$

$$i = 0 \dots N(k) + K - 1, \tag{8g}$$

$$j = \begin{cases} \lfloor i/L \rfloor & \text{if } i \le N(k) \\ \lfloor N(k)/L \rfloor & \text{if } i > N(k) \end{cases} \tag{8h}$$

$$\underline{v} \le v_\ell \le 1, \quad \ell = 0 \dots M(k) - 1 \tag{8i}$$

where $\tilde{x}$, $\tilde{A}_d$ and $\tilde{B}_d$ are obtained from x, A_d and B_d by removing the first state variable x_S. Variable ε is introduced to make constraint (8e) soft. A common solution of adding ε to the right hand side of $x_H^{(i)} \le \bar{x}_H$ does not work here, because this formulation cannot be transformed to a geometric programming constraint. Therefore ε is used as a scaling of $\bar{x}_H$ and $\varepsilon \ge 1$ is prescribed. The value of ε is minimized by completing the cost with an additive term $w_\varepsilon \varepsilon$ penalizing the constraint violation. By using the soft constraint, the numerical infeasibility caused by $x_H^{(i)} > \bar{x}_H$ can be avoided. After the optimization is completed, the value of ε gives information on the magnitude of the constraint violation that has been occurred during the synthesis.

This problem can be systematically transformed to a convex geometric program, which can be solved efficiently by numerical optimization. To perform the simulations presented in this paper we use CVX optimization package Grant and Boyd (2020) to formulate the geometric program. CVX also calls an external solver, which in our case is MOSEK MOSEK ApS (2020), to solve the convex optimization problem.

Remark 2. Further ingredients can be added to (8) to decrease the gap between the convex and the original optimization problems. Since the upper bound $\bar{x}_S$ is constant over the entire horizon, the simplified and the original model have similar behavior, if the trajectory of x_S does not significantly differ from $\bar{x}_S$. To keep $x_S - \bar{x}_S$ under control, the dynamics of $x_Z := 1 - x_S$ can be added to the model. Since the dynamics of x_Z in discrete time is polynomial with positive coefficients, thus the optimization problem can still be converted to a geometric program. By prescribing upper bound for x_Z, we can limit the changes of x_S. The drawback of this modification is that constraining x_S narrows the set of feasible control policies. To overcome this limitation an iterative design can be applied. After each iteration, the trajectory of x_S (computed from x_Z) is stored and used in the next iteration in the place of the constant $\bar{x}_S$. (The simulations for this paper (Section 5) have been performed without this fine tuning step, so the control input has been obtained in one step by solving the geometric optimization problem (8) only once.

Remark 3. Though in most cases the convex program above produces an acceptable result, it is possible to further improve the control policy by nonlinear optimization. For this, the control sequence generated by (8) can be used as a warm start for the nonlinear optimization (6).

4.3. Suboptimal solution by linear programming

We have seen that geometric programming provides an efficient way to simplify the optimization task required by the predictive control design. Now we go further. In this section we show that by applying additional relaxations it is possible to convert the original problem to a linear program (LP). LP is the simplest convex optimization task, it can be solved very efficiently even for thousands of free variables Gass (2003).

To generate a linear program from (6) the key step is choosing $\nu = [\nu^{(0)}, \nu^{(1)}, \dots \nu^{(N(k)-1)}]$ with definition $\nu^{(i)} = v^{(i)}(x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)})\bar{x}_S$ to be the new optimization variable. Here $\bar{x}_S$ is the constant initial value of x_S and it substitutes $x_S^{(i)}$, because we can assume that x_S does not change significantly during the control period. Clearly, the dynamics are linear in $\nu^{(i)}$. The input constraints $\underline{v} \le v^{(i)} \le 1$ can be implemented by prescribing $\underline{v}(x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)})\bar{x}_S \le \nu^{(i)} \le (x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)})\bar{x}_S$, which is again linear in all variables. After computing the sequence $\nu^{(0)}, \nu^{(1)}, \dots \nu^{(N(k)-1)}$ the control input can be obtained by $v^{(i)} = \nu^{(i)}/((x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)})\bar{x}_S)$.

This reformulation greatly simplifies the optimization, but generates also some new problems. One of the problems is that, we can no longer prescribe piecewise constant input by simple linear constraints.

This requirement should therefore be dropped from the optimization procedure and has to be handled by post processing the result. If the control input is allowed to change only at every $L \cdot T_s$ time step, the simplest way is to take the mean of the L $\upsilon^{(i)}$ values in the same $L \cdot T_s$ window and to consider the result as the constant input over this time interval. This 'approximation' gives acceptable result only if υ does not change significantly inside the $L \cdot T_s$ intervals. Unfortunately this does not hold in our case: the LP tends to generate highly oscillating input sequence at certain sections of the control window. The piecewise constant approximation of this signal results in significant performance loss and even infeasibility of the optimization in the later time steps. Since the epidemic dynamics is relatively slow, we have found that not the term $x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)}$ is responsible for the oscillation. Consequently, if ν is smoothed out, so can be expected from υ. We therefore applied the following strategy: ν is approximated by a 4th order B-spline The MathWorks (2020) and the optimization is reformulated to seek the coefficients of the spline instead of the $\nu^{(i)}$ values directly. To avoid large changes inside the $L \cdot T_s$ length sections, the knot points are a-priori fixed at the boundaries of these intervals. This prevents ν from oscillating between the successive knots. As the spline bases (the knot points) are a-priori fixed, the optimization remains linear in the parameters. Simulation results will show that this strategy works in practice: the spline formulation smoothes out the solution of the optimization and results in input sequences that can be suitably approximated by a piecewise constant function thereafter.

The second problem with LP relaxation is that the original cost is nonlinear in the optimization variable ν. Therefore it is replaced by maximizing $J_{LP} = \sum_{i=0}^{N(k)} w_i \nu^{(i)}$. This cost is of course, not equivalent to the original one, but again, it expresses similar intention: we seek the less stringent interventions that do not result in violation of state constraints. J_{LP} can be maximized by maximizing $\upsilon^{(i)}$-s (which is the original task) and the terms $x_P^{(i)} + x_I^{(i)} + \delta x_A^{(i)}$. Note that, these two requirements are not contradictory, because applying less stringent interventions induces the increase of the number of infected individuals, i.e. the values of x_I, x_A and x_P get larger. The increase is controlled by the upper bound prescribed for x_H.

To guarantee numerical feasibility the constraint $x_H \le \bar{x}_H$ is again transformed to a soft constraint $x_H \le \bar{x}_H + \varepsilon$, where $\varepsilon \ge 0$ is an additional free variable. Its value is minimized by adding $-w_\varepsilon \varepsilon$ to the linear cost above.

The control design by LP optimization can be summarized as follows:

$$\max_{\vartheta} \quad -w_\varepsilon \varepsilon + \sum_{i=0}^{N(k)-1} w_i \nu^{(i)} \tag{9a}$$

$$\text{w.r.t.} \tag{9b}$$

$$\nu^{(i)} = \begin{cases} \boldsymbol{\Phi}(iT_s)\vartheta & \text{if } i < N(k) \\ \nu^{(N(k)-1)} & \text{if } i \ge N(k) \end{cases} \tag{9c}$$

$$x^{(i+1)} = A_d x^{(i)} + B_d \nu^{(i)} \tag{9d}$$

$$x^0 = x_k \quad \text{(state at time } k\text{)}, \tag{9e}$$

$$x_H^{(i)} \le \bar{x}_H + \varepsilon \tag{9f}$$

$$0 \le \varepsilon \tag{9g}$$

$$i = 0 \dots N(k) + K - 1, \tag{9h}$$

$$\underline{\upsilon}(x_P^{(j)} + x_I^{(j)} + \delta x_A^{(j)})\bar{x}_S \le \nu^{(j)} \le (x_P^{(j)} + x_I^{(j)} + \delta x_A^{(j)})\bar{x}_S \tag{9i}$$

$$j = 0 \dots N(k) - 1 \tag{9j}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the row vector containing the values of B-spline basis functions evaluated at iT_s and ϑ is the vector of the coefficients. The spline is defined over the $[0, (N(k)-1)T_s]$ interval.

5. Output feedback control

In this section an output feedback controller is formed by using the state observer of Section 3 and the two state feedback MPC controllers of Section 4. The properties of the control structure are analyzed via numerical simulations. In these simulations, the sensitivity of the controller to the uncertainty of the model parameters is also examined.

5.1. Analysis of the observer

To analyze the proposed observer structure, numerical simulations have been performed. The observer is tested in open loop without controller. The control input is chosen to be constant 1, which corresponds to applying no intervention. The initial state is selected to represent the beginning of the epidemic, that is the number of infected persons is very low, there are no hospitalized patients yet and nobody has been recovered or died. Numerically $x_0 = [x_{0,S};\ 200;\ 50;\ 2;\ 1;\ 0;\ 0]/X$, where $x_{0,S} = X - 200 - 50 - 2 - 1$ and X is the total population, which is $X = 9800000$ in Hungary. The Kalman filter required to reconstruct x_P, x_I, x_A (Section 3.2) has been designed by assuming only a Gaussian output noise (i.e. a noise on x_H) with covariance $R_{cov} = 0.1$. In the simulations we assumed that the model parameters are not known exactly: the observer is designed for a nominal model and then applied to different "real" models generated by perturbing the nominal parameters. The parameters have been perturbed according to the following uncertainty bounds:

$$\begin{array}{|c|c|c|c|c|c|} \hline \alpha & \pm 20\% & p & \pm 5\% & \beta & \pm 10\% \\ \delta & \pm 10\% & q & \pm 20\% & \rho_I & \pm 10\%, \\ \rho_A & \pm 10\% & \eta & \pm 10\% & h & \pm 10\%, \\ \hline \end{array} \tag{10}$$

These bounds are based on the latest uncertainty estimates calculated by using the epidemiological data collected worldwide. More information on the identification and uncertainty bounds of the model parameters can be found e.g. in Fernández-Villaverde and Jones (2020), Korolev (2021).

Taking the uncertainty intervals above, 100 different parameter sets (i.e. 100 different models) have been generated randomly by using uniform distribution over the uncertainty intervals. The relative absolute estimation error (i.e. the error divided by the maximal value of the state) are depicted for all scenarios in Fig. 3. It is not surprising that in the uncontrolled case the epidemic reaches a peak where the number of infected individuals are extremely high. We have found that the estimation error is maximal in the neighborhood of the extremal values of the states. Since the goal of the control is to avoid (delay) the epidemic peak, therefore smaller estimation errors are expected in closed loop. This will be justified in the next section.

By analyzing Fig. 3, a small estimation error can be seen even in the nominal case. This can be explained by the effects of model reduction and the numerical integrations needed to compensate the time delay. Considering the uncertain cases, the relative error is the highest in the estimation of x_A, while at the other states it is less than 20%. This is not surprising as the model is nonlinear, several parameters are allowed to deviate from the nominal value and the observer is designed without explicitly taking the uncertainty into consideration. Nevertheless the observer well tracks the trend of the states and the accuracy it provides is sufficient to use the estimated states for feedback control design.

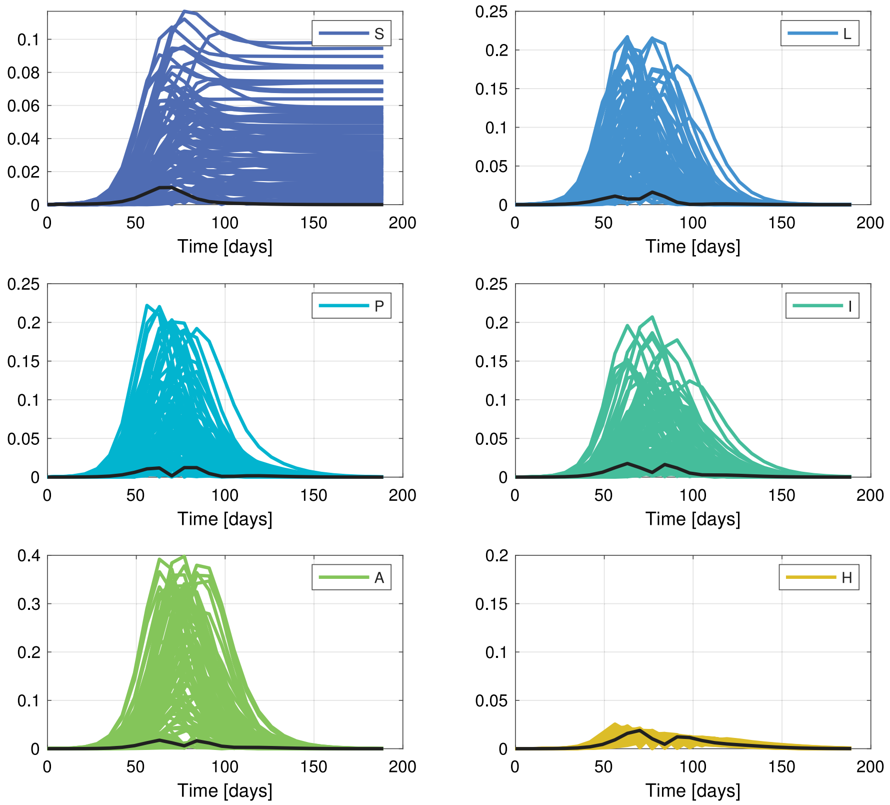

Fig. 3. Analysis of state reconstruction error by testing the observer on 100 randomly generated epidemic models. The figures show for each state the relative absolute estimation error, i.e. absolute error divided be the maximal value of the state. The nominal case is highlighted by black.

5.2. Closed loop control by geometric programming

In this section the MPC controller introduced in Section 4 is applied to our particular epidemic model. The controller uses the state estimation provided by the observer (Section 3), they thus form an output feedback structure. The parameters of the controller have been chosen as follows. The control window is 190 days, so $H = 190/T_s$. K is set to 3 weeks, so $K = 21/T_s$. The upper bound for the hospitalized patients (state x_H) is chosen to $\bar{x}_H = 0.001$. This bound is based on the available capacity of the healthcare system in Hungary. The system is assumed to start from the initial state (10), but the controller (and the observer) are activated only after 2 weeks. We assume that these two weeks are needed to detect the outbreak of an epidemic. The lower bound of the control input is set to $\underline{v} = 0.2$, which corresponds to the strictest measure, the total lockdown of the cities. This value was determined using Wang et al. (2020) studying the situation in Wuhan in 2019–2020.

In the simulations we have used 6 models: beside the nominal one, 5 others with different parameter sets have been chosen to represent uncertain scenarios. Actually, these 5 models have been obtained by selecting the 5 worst case scenarios from the 100 random examples used in the observer analysis in Section 5.1. The selection was based on the maximal estimation error. The controllers have been tested on the continuous time model, i.e. in each sampling instant the actual control input is applied on the continuous model.

We have performed two simulation scenarios. In the first case (simulation scenario I.) all of the weights w_i in the cost function are set to 1, i.e. no weighting is applied. In the second case (simulation scenario II.), we have chosen the weights to reflect the tolerance of the society for the interventions. In the first four weeks the weight is 1, which means that strict measures are tolerated. In the next four weeks the control input weights are 4 to force less stringent interventions. From the 9th week the weight is 12, that means strict interventions are definitely undesired. In compact form, the initial weight sequence at $k = 0$ is $w_{0,\ldots,3} = 1$, $w_{4,\ldots,15} = 4$, $w_{16,\ldots,M(0)-2} = 12$, $w_{M(0)-1} = 1$. As time index k gets closer to H, $M(k)$ decreases (shrinking horizon), the actual weights are obtained by taking the last $M(k)$ elements of the initial weight sequence above. The small weight of the last control action is part of the terminal constraints (see Section 4 for details). The weight w_ε penalizing the soft constraint violation has been set to 10.

Figs. 4, 5, 6 and 7, 8, 9 present the state trajectories and control inputs obtained in the two simulation scenarios. The nominal case is distinguished from the uncertain ones. It can be seen that in the nominal case, the safety ingredients guarantee the constraint satisfaction over the entire horizon, as well as beyond the control window. On the other hand if the model parameters significantly differ from the nominal ones, x_H may slightly exceed the limit after the end of the control window. This motivates the design of a robust controller, e.g. by a simple scenario method introduced in the next subsection.

It can be seen that the two different weightings result in two different control strategies. In the first case (when $w_i = 1$, $\forall i$), an almost constant, mild intervention is proposed, which is gradually eased towards the end of the control window. In the second case, the control input follows the same pattern as the weighting: in the first 4 weeks strict intervention is applied, that is significantly eased in the next month. From the 8th week, the control input is pushed as close as possible to 1.

Finally, we analyze the computation time needed to solve the optimization task (8). The computation consists of two main steps: first, formulating the GP problem (CVX) and second, finding the optimal solution (MOSEK). The first part requires about 180–200 s while the second takes 70–80 s in general (measured on the longest control horizon, at $t = 0$). (The computations have been performed on Mac Pro Late 2013 computer, under Matlab 2020a.) Compared to the nonlinear MPC methods in Péni et al. (2020), where finding an optimal solution required 10–12 min, this is a significant improvement in terms of

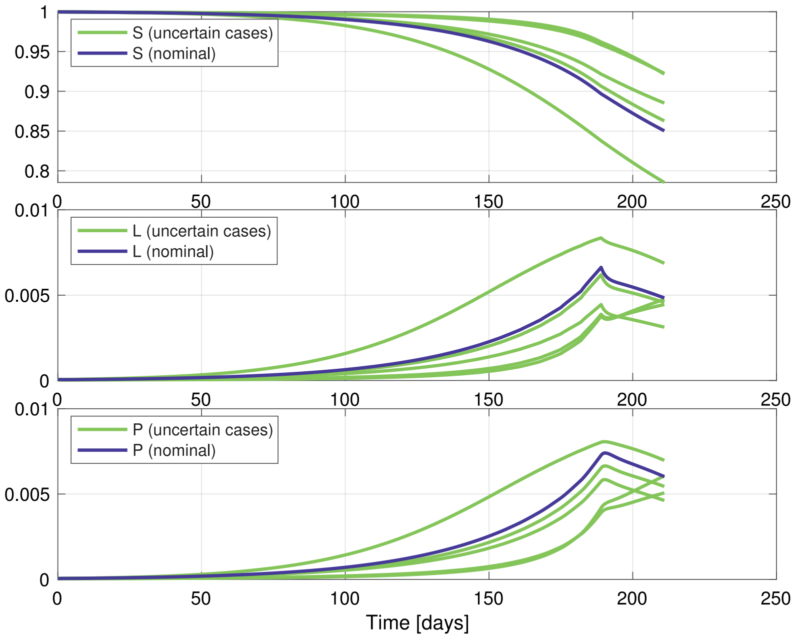

Fig. 4. Trajectories of states x_S, x_L, x_P obtained in closed-loop simulation by using Geometric Programming based relaxation. All of the weights in the cost function are 1 (simulation scenario I).

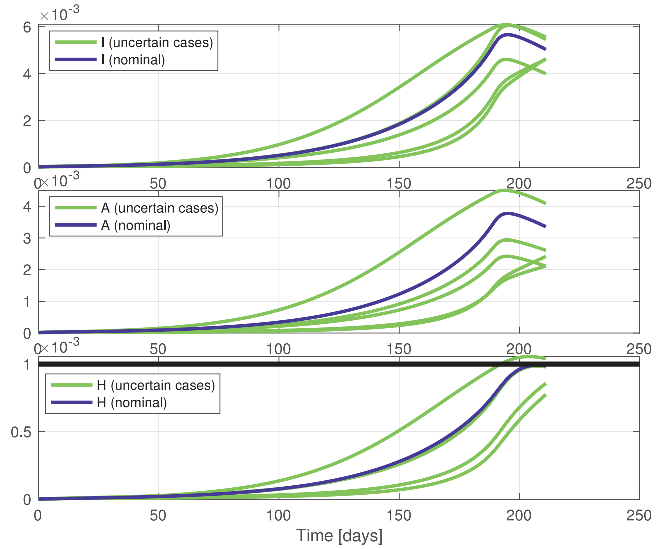

Fig. 5. Trajectories of states x_I, x_A, x_H obtained in closed-loop simulation by using Geometric Programming based relaxation. All of the weights in the cost function are 1 (simulation scenario I). The upper limit $\bar{x}_H$ is indicated by solid black line.

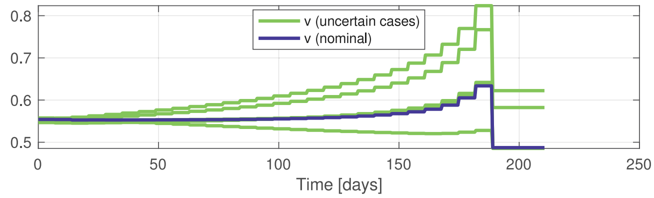

Fig. 6. Control input v obtained in closed-loop simulation by using Geometric Programming based relaxation. All of the weights in the cost function are 1 (simulation scenario I).

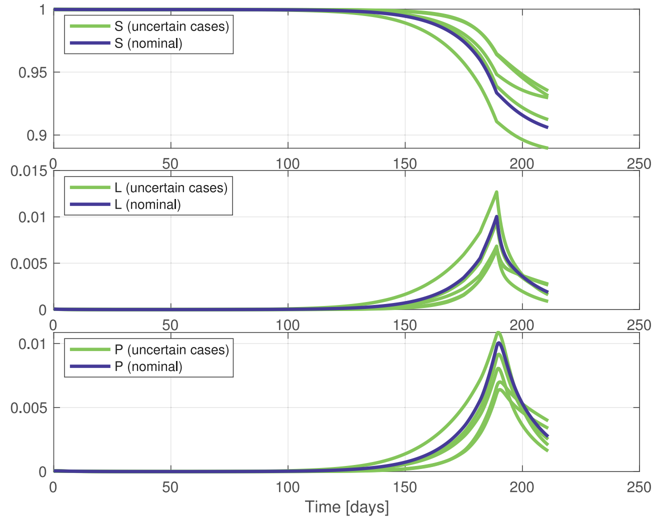

Fig. 7. Trajectories of states x_S, x_L, x_P obtained in closed-loop simulation by using Geometric Programming based relaxation. The weights in the cost function are chosen to reflect the tolerance of the society for the interventions (simulation scenario II).

computational cost. However, it can be seen that the problem formulation takes up significant time compared to the numerical optimization. Since every time a similar numerical optimization task has to be performed (only the length of the horizon changes), it is possible to speed up the problem formulation phase by generating precompiled program components, identifying tasks that have to be performed only once and exploiting the specific structure of the problem. With these improvements the computation time can be further reduced.

5.3. Closed loop control by linear programming and simple scenario method

This section presents the results obtained by using the LP based control design presented in Section 4.3. To formulate and solve the LP optimization problem we used again the CVX environment. As we have mentioned, the main drawback of CVX is that the optimization problem has to be built up every time its solution is needed. In case of LP, CVX proposes a tool, called CVXGEN Mattingley and Boyd (2011)

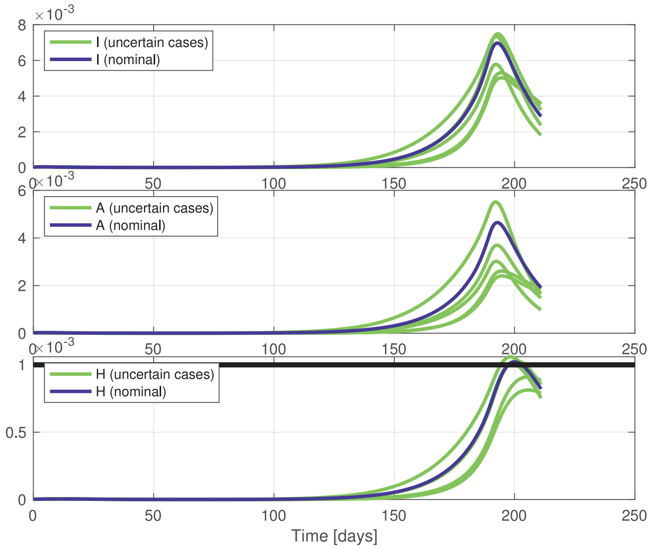

Fig. 8. Trajectories of states x_I, x_A, x_H obtained in closed-loop simulation by using Geometric Programming based relaxation. The weights in the cost function are chosen to reflect the tolerance of the society for the interventions (simulation scenario II). The upper limit $\bar{x}_H$ is indicated by solid black line.

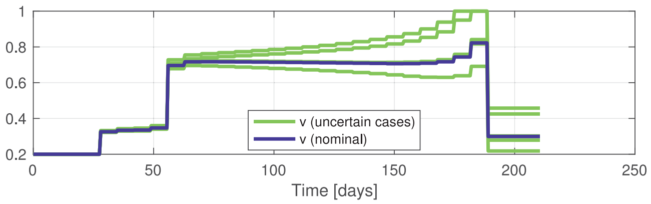

Fig. 9. Control input v computed in closed loop simulations by using Geometric Programming based relaxation. The weights in the cost function are chosen to reflect the tolerance of the society for the interventions (simulation scenario II).

which can compile the code and allows to generate parameterized optimizer objects. In this paper, we do not use this tool as we have found standard CVX suitably fast. However, by exploiting the benefit of code optimization and precompiling, the computation time can be further decreased.

By performing a control synthesis with the same parameters as in the GP example above (i.e. $N = 190/T_s$, $K = 21/T_s$, $L = 7/T_s$), the computation time with LP is less than 10 s. From this, the construction of the optimization task takes approximately 8–8.5 s, the solve-time is always below 2 s. Since the solution of an LP requires significantly less time than GP, therefore we implemented a simple scenario method to improve the robustness of the controller. This approach can be applied to the GP-based control synthesis as well.

The basic concept of scenario based predictive control is generating a pool of possible system models and designing a control input that is feasible (and suboptimal) for all. If the size of the pool is suitably large, the probability of the constraint violation (infeasibility) on an arbitrary new model is close to zero. The details of scenario methods and scenario based control design can be found e.g. in Calafiore and Fagiano (2013), Campi (2018). The main drawback of this approach is that the computation of the control input generally requires to include the entire pool in the optimization process, which highly increases the dimension of the problem. Due to the specific structure of the epidemic model, this difficulty is not to be reckoned with. Since smaller control inputs mean stricter interventions, if feasible control inputs are computed separately for the models of the pool, then the minimum of these control input candidates gives a feasible control action for all models. The papers cited above propose specific algorithms on how to set up the pool correctly. In this paper we chose a very simple method. The pool is initiated with the nominal model. Random models are generated and the scenario MPC design is applied. If a constraint violation is detected, the actual model is added to the pool. The pool for the simulations presented in this paper has been constructed by running 100 design processes with 100 different models. (In all cases the weights of the cost function have been set to 1 (no input weighting has been applied), the weight w_ε of constraint violation has been set to 100.) After completing all of the design tasks we have obtained a pool of only 7 models. By testing the pool on another 100 examples, we have not run into constraint violation. This result suggests that the scenario method is a potential approach to design robust controller for our epidemic model. This motivates us to continue the analysis of this method and to make further efforts to derive mathematical guarantees for feasibility and optimality. This will be part of the future research.

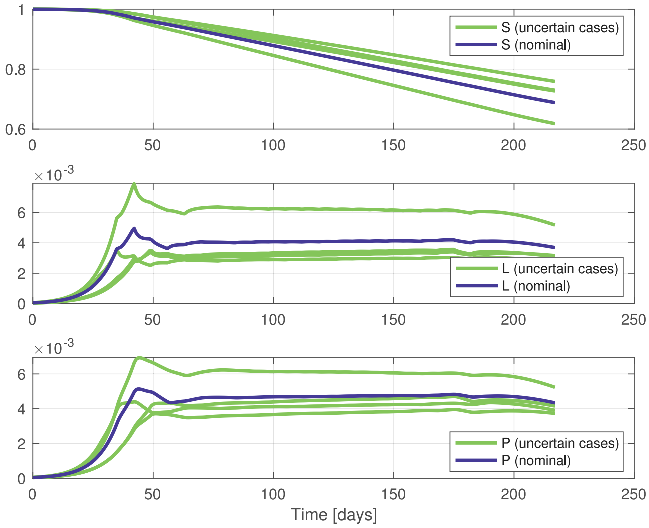

Fig. 10. Trajectories of states x_S, x_L, x_P obtained in closed-loop simulation by using Linear Programming based design. All of the weights in the cost function are 1.

After constructing the pool of models, we have tested the scenario controller in closed loop on the same 6 models that have been used in the earlier simulations. As before, we have not applied any weighting in the cost, i.e. $w_i = 1$ has been chosen in (9) for all i. The weight w_ε penalizing the soft constraint violation has been set to 100. The simulation results can be seen in Figs. 10–12. Similar to the results obtained by the GP-based design (simulation scenario I.), a mild intervention is applied in the most part of the control window, which is gradually relaxed towards the end of the horizon. The main difference between the two control strategies is at the beginning of the control window: the LP strategy starts with very mild measures, the control is practically activated only after the 4th week. In contrast, the GP-based approach applies significant interventions right from the beginning. By comparing the cost of the control we have computed the L_1 norm of the control signals in both cases: by using the GP-based synthesis this norm is between 105 and 125, while it is between 115 and 135 in case of the LP. The LP is slightly better, but the difference is not significant, so we can say that practically the performance of the two control strategies is very similar. Note also that the characteristic and performance of either control strategy can be further tuned by modifying the weights w_i of the cost function.

By analyzing the trajectories of x_H we can see that with the LP controller there is no constraint violation. Due to the scenario design, a slightly conservative control action is designed, which prevents x_H from reaching the prescribed limit.

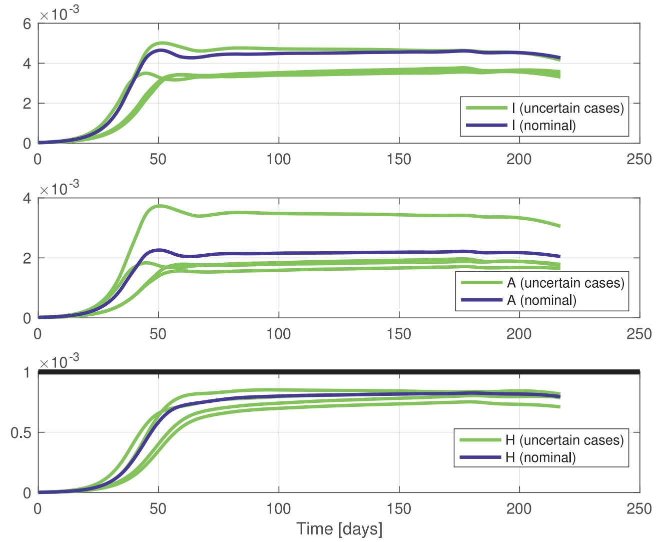

Fig. 11. Trajectories of states x_I, x_A, x_H obtained in closed-loop simulation by using Linear Programming based design. All of the weights in the cost function are 1. The upper limit $\bar{x}_H$ is indicated by solid black line.

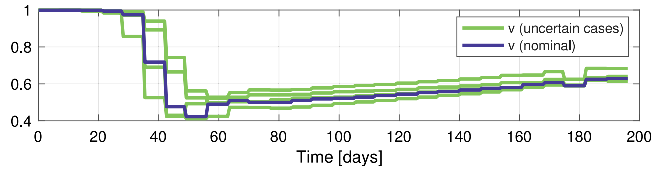

Fig. 12. Control input v computed in closed-loop simulations by using Linear Programming based design. All of the weights in the cost function are 1.

6. Conclusions

An output feedback control approach was proposed in this paper to design the stringency of the non-pharmaceutical measures for mitigating the COVID-19 pandemic. The linearity of a subsystem of the used compartmental model allowed us to use dynamic inversion to give an estimation for the number of people in the latent compartment. The nonnegativity of the terms in the discrete-time model was exploited to formulate the control problem in a convex geometric programming framework. Additionally, by relaxing the optimization task further, the input can be computed by solving a simple linear programming problem. A state observer was designed and analyzed for the tracking of non-measured variables. The maximal estimation errors occur around the peak of the epidemic. Simulation results show that the designed feedback efficiently reduces the effects of the studied model uncertainties, and the control goals and constraints can be reached within an acceptable tolerance even in the linear programming case. The computations can be done in real-time and significantly faster than in the case of a previous nonlinear MPC solution Péni et al. (2020). The convexity of the re-formulated optimization problems guarantees a unique solution which is suboptimal with respect to the original optimal control problem, and makes possible to extend the control design to larger dimensional (e.g. multi-scale or multi-region) compartmental models Carli et al. (2020), as well as to apply more advanced uncertainty analysis techniques, e.g. scenario methods, in the future. Another subject of further work will be the application of the obtained suboptimal inputs as initial solutions in the original nonlinear MPC design.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Anderson, R. M., & May, R. M. (1992). *Infectious diseases of humans: Dynamics and control*. Oxford University Press.

Ansumali, S., Kaushal, S., Kumar, A., Prakash, M. K., & Vidyasagar, M. (2020). Modelling a pandemic with asymptomatic patients, impact of lockdown and herd immunity, with applications to SARS-CoV-2. *Annual Reviews in Control*, *50*, 432–447. http://dx.doi.org/10.1016/j.arcontrol.2020.10.003.

Bhattacharyya, S. (1978). Observer design for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, *23*(3), 483–484.

Boyd, S., Kim, S.-J., Vandenberghe, L., & Hassibi, A. (2007). A tutorial on geometric programming. *Optimization and Engineering*, *8*(1), 67–127. http://dx.doi.org/10.1007/s11081-007-9001-7.

Boyd, S., & Vandenberghe, L. (2009). *Convex optimization*. Cambridge University Press, URL https://web.stanford.edu/~boyd/cvxbook/.

Brauer, F., Castillo-Chavez, C., & Castillo-Chavez, C. (2012). *Mathematical models in population biology and epidemiology (Vol. 2)*. Springer.

Calafiore, G. C., & Fagiano, L. (2013). Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, *58*(1), 219–224. http://dx.doi.org/10.1109/tac.2012.2203054.

Campi, M. (2018). *Introduction to the scenario approach*. Philadelphia: Society for Industrial and Applied Mathematics Mathematical Optimization Society.

Carli, R., Cavone, G., Epicoco, N., Scarabaggio, P., & Dotoli, M. (2020). Model predictive control to mitigate the COVID-19 outbreak in a multi-region scenario. *Annual Reviews in Control*, *50*, 373–393. http://dx.doi.org/10.1016/j.arcontrol.2020.09.005.

Fernández-Villaverde, J., & Jones, C. I. (2020). *Estimating and simulating a SIRD model of COVID-19 for many countries, states, and cities: Tech. rep.*, National Bureau of Economic Research.

Gass, S. I. (2003). *Linear programming: Methods and applications*. Courier Corporation.

Grant, M., & Boyd, S. (2020). CVX: Matlab software for disciplined convex programming, version 2.2. http://cvxr.com/cvx.

Hayhoe, M., Barreras, F., & Preciado, V. M. (2020). Data-driven control of the COVID-19 outbreak via non-pharmaceutical interventions: A geometric programming approach. ArXiv preprint arXiv:2011.01392.

Köhler, J., Schwenkel, L., Koch, A., Berberich, J., Pauli, P., & Allgöwer, F. (2020). Robust and optimal predictive control of the COVID-19 outbreak. *Annual Reviews in Control*, *51*, 525–539. http://dx.doi.org/10.1016/j.arcontrol.2020.11.002.

Korolev, I. (2021). Identification and estimation of the SEIRD epidemic model for COVID-19. *Journal of Econometrics*, *220*(1), 63–85. http://dx.doi.org/10.1016/j.jeconom.2020.07.038.

Mandel, A., & Veetil, V. (2020). The economic cost of COVID lockdowns: An out-of-equilibrium analysis. *Economics of Disasters and Climate Change*, *4*(3), 431–451.

Massonis, G., Banga, J. R., & Villaverde, A. F. (2020). Structural identifiability and observability of compartmental models of the COVID-19 pandemic. *Annual Reviews in Control*, *51*, 441–459. http://dx.doi.org/10.1016/j.arcontrol.2020.12.001.

Mattingley, J., & Boyd, S. (2011). CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, *13*(1), 1–27. http://dx.doi.org/10.1007/s11081-011-9176-9.

Morato, M. M., Bastos, S. B., Cajueiro, D. O., & Normey-Rico, J. E. (2020). An optimal predictive control strategy for COVID-19 (SARS-CoV-2) social distancing policies in Brazil. *Annual Reviews in Control*, *50*, 417–431. http://dx.doi.org/10.1016/j.arcontrol.2020.07.001.

Morato, M. M., Pataro, I. M., da Costa, M. V. A., & Normey-Rico, J. E. (2020). A parametrized nonlinear predictive control strategy for relaxing COVID-19 social distancing measures in Brazil. *ISA Transactions* (in press). http://dx.doi.org/10.1016/j.isatra.2020.12.012.

MOSEK ApS (2020). The MOSEK optimization toolbox for MATLAB 9.2.32. https://www.mosek.com.

Nowzari, C., Preciado, V. M., & Pappas, G. J. (2016). Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, *36*(1), 26–46. http://dx.doi.org/10.1109/MCS.2015.2495000.

Péni, T., Csutak, B., Szederkényi, G., & Röst, G. (2020). Nonlinear model predictive control with logic constraints for COVID-19 management. *Nonlinear Dynamics*, *102*(4), 1965–1986. http://dx.doi.org/10.1007/s11071-020-05980-1.

Phipps, S. J., Grafton, R. Q., & Kompas, T. (2020). Robust estimates of the true (population) infection rate for COVID-19: A backcasting approach. *Royal Society Open Science*, *7*(11), Article 200909. http://dx.doi.org/10.1098/rsos.200909.

Röst, G., Bartha, F. A., Bogya, N., Boldog, P., Dénes, A., Ferenci, T., et al. (2020). Early phase of the COVID-19 outbreak in Hungary and post-lockdown scenarios. *Viruses*, *12*(7), 708. http://dx.doi.org/10.3390/v12070708.

Saladino, V., Algeri, D., & Auriemma, V. (2020). The psychological and social impact of COVID-19: new perspectives of well-being. *Frontiers in Psychology*, *11*, 2550. http://dx.doi.org/10.3389/fpsyg.2020.577684.

The MathWorks (2020). Spline toolbox for use with MATLAB, release 2020a.

Villaverde, A. F., Barreiro, A., & Papachristodoulou, A. (2016). Structural identifiability of dynamic systems biology models. *PLoS Computational Biology*, *12*(10), Article e1005153.

Wang, H., Wang, Z., Dong, Y., Chang, R., Xu, C., Yu, X., et al. (2020). Phase-adjusted estimation of the number of coronavirus disease 2019 cases in Wuhan, China. *Cell Discovery*, *6*(1), 1–8.