This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



# A Scenario-Based Stochastic MPC Approach for Problems With Normal and Rare Operations With an Application to Rivers

Hasan Arshad Nasir, Algo Carè, and Erik Weyer

***Abstract*— This paper formulates a control problem for systems that are affected by uncertain inputs and are vulnerable to risks as a chance-constrained optimization problem (CCP) with two chance-constraints (CCs). The first CC encompasses requirements of the normal operations of the system, whereas the second CC ensures the avoidance of risks associated with rare events. CCPs are in general difficult to solve, and this paper proposes a scenario-based *optimization, testing, and improving algorithm* to find approximate solutions to such problems within a stochastic model predictive control setting in a computationally cheap manner. The proposed approach is applied to a river control problem with flood avoidance, and the controller performed well in realistic simulations of the upper part of Murray River in Australia.**

***Index Terms*— Chance-constrained optimization problem (CCP), risk mitigation, scenario approach, stochastic model predictive control (MPC).**

## I. Introduction

In ALL real systems, there are physical and operational constraints, which must be considered when controllers are designed. Model predictive control (MPC) [1] is often a natural choice in such situations because of its ability to explicitly consider constraints in the problem formulation.

Some systems are affected by disturbances, which are difficult to bound *a priori*. In such cases, hard robust constraints in the formulation of the control problem may not only degrade the performance of the solution due to conservativeness, but they can also lead to an infeasible problem. For such systems, probabilistic constraints are useful. The constraints are now only required to be satisfied with a certain probability, and this leads to a chance-constrained optimization problem (CCP) [2]–[5]. An optimization problem with multiple chance-constraints (CCs), each to be satisfied with different probabilities, is known as a multiple CCP (M-CCP) [6]. In this paper, we propose an algorithm to find approximate solutions to M-CCPs with two CCs within a stochastic MPC setting.

A problem which naturally leads to an M-CCP formulation, and which is the main motivation for this paper, is control of rivers [7], [8]. Rivers have unregulated in-flows from tributaries which are uncertain, and using deterministic measures to bound their values is neither easy nor natural. Under normal circumstances, water levels, flows and possibly their rates of change should be kept within certain limits to ensure satisfactory environmental outcomes and a certain level of service to irrigators and other users. The limits can, however, occasionally be exceeded by small amounts without causing any major problems, and hence imposing a probabilistic constraint is natural. In addition, there are (higher) water level limits that should not be exceeded, as it would lead to flooding and major damage, and such water level constraints should be satisfied with a very high probability. In practice, when there is a risk of flooding, river operators switch the operational mode, and flood avoidance becomes the main objective.

In this paper, we consider systems with two modes of control operations: normal operations and operations for rare situations (e.g., flood avoidance in rivers). Such a control problem is formulated as a CCP with two CCs, where the first CC is related to normal operations and the second is related to risk mitigation.

Chance-constrained problems are generally nonconvex and difficult to solve. Randomized strategies [9]–[17] provide computationally tractable approximate solutions to such problems, especially the scenario approach introduced in [9]–[12], [17], and [18] is promising because of its simplicity. It does not require any specific assumption on the nature of the disturbance, and it can also be extended to M-CCPs [6], [19]. However, due to the high probability with which the second CC needs to be satisfied, the corresponding scenario problem can become computationally very expensive.

In this paper, we propose an *optimization, testing, and improving (OTI) algorithm* that uses the scenario approach together with ideas borrowed from validation set methods [20]–[24] to find approximate solutions to CCPs with two CCs. The algorithm is motivated by applications where the CCs possess a certain form of nestedness, where the satisfaction of constraints associated with normal operations usually,

Manuscript received September 4, 2017; revised January 22, 2018; accepted February 8, 2018. Manuscript received in final form February 27, 2018. The work of H. A. Nasir was supported by National ICT Australia. The work of A. Carè was supported in part by the ARC Discovery Project under Grant DP130104028 and in part by the ERCIM Alain Bensoussan Fellowship Program. The contribution of A. Carè was carried out partly while he was a Research Fellow at the University of Melbourne. Recommended by Associate Editor L. Fagiano. *(Corresponding author: Hasan Arshad Nasir.)*

H. A. Nasir and E. Weyer are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: hanasir@unimelb.edu.au; ewey@unimelb.edu.au).

A. Carè is with the Department of Information Engineering, University of Brescia, 25123 Brescia, Italy (e-mail: algo.care@unibs.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2018.2811404

but not always, leads to satisfaction of the risk avoidance constraints. For example, in rivers, if the water level stays below the limit associated with normal operations, it also stays below the limit associated with flood avoidance. Nonetheless, the probability of a flood event must be constrained to be much smaller than the probability that a normal operation constraint is violated. Therefore, the CC related to normal operations does not imply the CC related to flood avoidance. The proposed algorithm is in three steps: *optimization*, *testing*, and *improving*. In the optimization step, the algorithm solves an optimization problem with the CC associated with normal operations only. In the testing step, which is computationally inexpensive, the algorithm tests the solution against the CC associated with risk mitigation. The found solution is accepted if the test is passed; otherwise, the solution is improved in a computationally cheap way by solving a 1-D scenario problem to satisfy the second CC in the improving phase.

In the proposed algorithm, the concept of a *default solution* plays a crucial role in the improving phase. A default solution is a possible backup solution, which in terms of the objective criterion may give poor performance, but it is the safest option when a situation takes a turn for the worse, e.g., close the gates completely when there is a high risk of flooding. This can give poor performance, since minimum flow conditions in parts of the river may be violated and the daily flow change may be too large. However, it is the safest option in terms of preventing flooding. In the improving procedure, the solution obtained from the optimization step is moved closer to the default solution. This is achieved by solving a 1-D scenario problem with the only aim of satisfying the CC associated with risk mitigation.

The main difference between the OTI algorithm and sequential randomized algorithms, such as those in [21]–[23], is that, in our scheme, the most informative sampled constraints in the testing step (namely, the constraints that violated the solution obtained in the optimization step) are reused in the improving phase.[1] This helps in reaching low violation probabilities using a small sample in the improving step. Moreover, our scheme deals with two different kinds of CCs and treats them differently, since they serve different purposes for the problem at hand. While the proposed algorithm is applicable to general systems where the above described properties (namely, the nestedness of CCs and the availability of a default solution) are satisfied, the main motivation and contribution of this paper is in the area of river control.

This paper is organized as follows. In Section II, we formulate the control problem and state the two CCs. A discussion of some existing techniques for solving M-CCPs is given in Section III. The OTI algorithm is introduced in Section IV. The river control problem is presented in Section V, and the proposed algorithm is applied to operational data from the upper part of Murray River in Australia. Conclusions are given in Section VI.

[1] This idea is inspired by the approach in [24] [see Section III-C (a comparison with validation methods) therein]. In [24], however, the existence of a robustly feasible solution is assumed, while in this paper, we relax this assumption by introducing the concept of a default solution and by adding suitable tests.

## II. Problem Formulation

In this section, we give the system description and formulate the control problem.

### A. System Description

We consider the following state space system:

$$x_{n+1} = Ax_n + Bu_n + w_n \tag{1}$$

$$y_n = Cx_n \tag{2}$$

where $x \in \mathbb{R}^{n_s}$ is the state vector, $u \in U \subset \mathbb{R}^m$ is the input, $w \in W \subset \mathbb{R}^{n_s}$ is a vector of disturbances (uncertain variables), and $y \in \mathbb{R}^p$ is the output.

Over a finite time horizon $M$, let the vectors of states, inputs, disturbances and outputs be $\mathbf{x}_{n+1} = \begin{bmatrix}x_{n+1} & x_{n+2} & \dots & x_{n+M}\end{bmatrix}^\mathsf{T}$, $\mathbf{u}_n = \begin{bmatrix}u_n & u_{n+1} & \dots & u_{n+M-1}\end{bmatrix}^\mathsf{T}$, $\mathbf{w}_n = \begin{bmatrix}w_n & w_{n+1} & \dots & w_{n+M-1}\end{bmatrix}^\mathsf{T}$, and $\mathbf{y}_n = \begin{bmatrix}y_n & y_{n+1} & \dots & y_{n+M-1}\end{bmatrix}^\mathsf{T}$, respectively. From (1), the relationship between the states at time $n$ and $n+i$ is given by

$$x_{n+i} = A^i x_n + \begin{bmatrix}A^{i-1}B & \dots & B\end{bmatrix}\begin{bmatrix}u_n \\ \vdots \\ u_{n+i-1}\end{bmatrix} + \begin{bmatrix}A^{i-1} & \dots & I\end{bmatrix}\begin{bmatrix}w_n \\ \vdots \\ w_{n+i-1}\end{bmatrix}. \tag{3}$$

Using (3) with $i = 1, 2, \dots, M$, we obtain the following compact model for $\mathbf{x}_{n+1}$, which will be used in the problem formulation:

$$\mathbf{x}_{n+1} = Fx_n + G\mathbf{u}_n + H\mathbf{w}_n \tag{4}$$

with

$$F = \underbrace{\begin{bmatrix}A \\ A^2 \\ \vdots \\ A^M\end{bmatrix}}_{(Mn_s \times n_s)}, \quad G = \underbrace{\begin{bmatrix}B & \mathbf{0} & \dots & \mathbf{0} \\ AB & B & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A^{M-1}B & A^{M-2}B & \dots & B\end{bmatrix}}_{(Mn_s \times Mm)}$$

$$H = \underbrace{\begin{bmatrix}I & \mathbf{0} & \dots & \mathbf{0} \\ A & I & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A^{M-1} & A^{M-2} & \dots & I\end{bmatrix}}_{(Mn_s \times Mn_s)}.$$

### B. Feedback-Based Control Policy

For systems with disturbances, it is advisable to optimize the objective function over feedback control policies rather than control values [25]. A natural choice is a time-varying state feedback control policy, i.e., $u_n = K_n x_n$, but optimization over $K_n$ is likely to be difficult, since the parametrisation is nonconvex in $K_n$ [26], [27].

However, the control policies can also be parametrized as affine functions of past disturbances [26], [27], which are linearly related to the current state [see (1) and (4)]. By using (1), $u_n = K_n x_n$ can be written as

$$u_n = K_n \prod_{i=0}^{n-1}(A + BK_i)x_0 + \sum_{j=1}^{n-1} K_n \prod_{i=j}^{n-1}(A + BK_i)w_{j-1} + K_n w_{n-1}. \quad (5)$$

Such a control policy can be parametrized as

$$u_n = \gamma_n + \theta_{n,0}w_0 + \theta_{n,1}w_1 + \dots \theta_{n,n-1}w_{n-1} \quad (6)$$

where $\theta_{n,i} \in \mathbb{R}^{m\times n_s}$ and $\gamma_n \in \mathbb{R}^m$. Over a finite horizon $M$, we have

$$\mathbf{u}_n = \Theta_n \mathbf{w}_n + \Gamma_n \quad (7)$$

where

$$\Gamma_n = \underbrace{\begin{bmatrix}\gamma_n^\top & \gamma_{n+1}^\top & \dots & \gamma_{n+M-1}^\top\end{bmatrix}^\top}_{(Mm\times 1)}$$

$$\Theta_n = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \theta_{n+1,n} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n+M-1,n} & \theta_{n+M-1,n+1} & \dots & \mathbf{0} \end{bmatrix}}_{(Mm\times Mn_s)}. \quad (8)$$

This parametrization is convex with respect to $\Theta_n$ and $\Gamma_n$ (for details see [26]–[28]). The parametrization in (8) also ensures causality of the control action. To use (7) as a control policy, the value of $w_{n-1}$ is needed at time $n$ [see (6)], and it can be obtained from

$$w_{n-1} = x_n - Ax_{n-1} - Bu_{n-1} \quad (9)$$

which assumes that the full state vector is measured.

### C. Stochastic MPC-Based Optimization Problem

In this section, we first introduce the two CCs related to normal operations and risk mitigation, respectively, and then we state the M-CCP problem to be solved in a stochastic MPC setting.

*1) Chance Constraint Related to Normal Operations:* This CC is given by

$$\mathbb{P}\{\mathbf{w}_n \in \mathbb{W} : \mathbf{u}_n(\mathbf{w}_n) \in \mathcal{U} \;\cap\; f(\mathbf{u}_n, \mathbf{w}_n) \le 0\} \ge 1 - \epsilon \quad (10)$$

where $\mathbb{P}$ is the probability according to which $\mathbf{w}_n$ is distributed, $\mathbb{W} = W^M$, $\mathbf{u}_n(\mathbf{w}_n)$ is the control action computed by the control policy given by (7), $\mathcal{U} = U^M$, $f(\mathbf{u}_n, \mathbf{w}_n)$ is a function, and $\epsilon \in (0, 1)$ is the allowed violation probability.

*2) Chance Constraint Related to Risk Mitigation:* This CC is given by

$$\mathbb{P}\{\mathbf{w}_n \in \mathbb{W} : g(\mathbf{u}_n, \mathbf{w}_n) \le 0\} \ge 1 - \epsilon_V \quad (11)$$

where $g(\mathbf{u}_n, \mathbf{w}_n)$ is a function and $\epsilon_V \in (0, 1)$ is the allowed violation probability with $\epsilon_V \ll \epsilon$.

The problem formulation and the algorithm presented ahead are motivated by applications, where the two CCs (10) and (11) possess a certain form of nestedness and involve the same decision variables. Instead of giving a formal definition of nestedness, we exemplify the concept by referring to the river example: we say that the CC related to normal operations (10) and the CC related to flood avoidance (11) are nested, because the constraints $f(\mathbf{u}_n, \mathbf{w}_n) \le 0$ and $g(\mathbf{u}_n, \mathbf{w}_n) \le 0$ are both constraints on the water level, and it holds that $g(\mathbf{u}_n, \mathbf{w}_n) \le f(\mathbf{u}_n, \mathbf{w}_n)$ for every $\mathbf{u}_n$ and $\mathbf{w}_n$. This means that if the water level stays below the limit associated with normal operations, it also stays below the limit associated with flood avoidance. Note, however, that although $f(\mathbf{u}_n, \mathbf{w}_n) \le 0$ implies $g(\mathbf{u}_n, \mathbf{w}_n) \le 0$, it is not true that one CC implies the other. In fact, the violation probability requirement for flood avoidance is higher than that of normal operations, i.e., $\epsilon_V \ll \epsilon$, as mentioned above.

The objective function $J(\Gamma_n, \Theta_n)$ for normal control operations is given by

$$J(\Gamma_n, \Theta_n) = \mathbb{E}\big[(\mathbf{x}_n - \mathbf{r})^\top \mathbf{Q}(\mathbf{x}_n - \mathbf{r}) + \mathbf{u}_n^\top \mathbf{R}\mathbf{u}_n + \Delta\mathbf{u}_n^\top \mathbf{S}\Delta\mathbf{u}_n\big] \quad (12)$$

where $\Delta\mathbf{u}_n = \begin{bmatrix}u_n \dots u_{n+M-1}\end{bmatrix}^\top - \begin{bmatrix}u_{n-1} \dots u_{n+M-2}\end{bmatrix}^\top$. We seek to minimize the following:

1) deviation of states $\mathbf{x}_n$ from a reference vector $\mathbf{r}$;
2) control actions $\mathbf{u}_n$ to avoid waste of system resources;
3) change in control actions $\Delta\mathbf{u}_n$.

$\mathbf{Q}$ and $\mathbf{R}$ are block diagonal matrices of positive definite weighting matrices, and $\mathbf{S}$ is a block diagonal matrix of positive semidefinite weighting matrices.

We use the objective function in (12) and the constraints in (4), (7), (10), and (11), and state the chance-constrained optimization problem as

$$\begin{aligned} \min_{\Gamma_n,\Theta_n}.\; & J(\Gamma_n, \Theta_n) \\ \text{s.t. } & \mathbb{P}\{\mathbf{w}_n \in \mathbb{W} : \mathbf{u}_n(\mathbf{w}_n) \in \mathcal{U} \;\cap\; f(\mathbf{u}_n, \mathbf{w}_n) \le 0\} \ge 1 - \epsilon, \\ & \mathbb{P}\{\mathbf{w}_n \in \mathbb{W} : g(\mathbf{u}_n, \mathbf{w}_n) \le 0\} \ge 1 - \epsilon_V, \\ & \mathbf{x}_{n+1} = Fx_n + G\mathbf{u}_n + H\mathbf{w}_n, \quad \mathbf{u}_n = \Theta_n\mathbf{w}_n + \Gamma_n \end{aligned} \quad (13)$$

where $\Gamma_n$ and $\Theta_n$ are the parameter matrices [see (7) and (8)]. We make the following assumptions.

1) *Set convexity:* $\mathcal{U}$ is a convex set.[2]
2) *Function convexity:* The functions $f(\mathbf{u}_n, \mathbf{w}_n)$ and $g(\mathbf{u}_n, \mathbf{w}_n)$ are convex[3] with respect to $\Gamma_n$ and $\Theta_n$.
3) *Default solution existence:* A default solution exists as detailed in the following.

*Default Solution:* The concept of the default solution is motivated from the river control problem, where, in order to avoid flooding, dam gates can be opened or closed completely. This is, however, an action that is not allowed during normal operations. We denote the default solution as $\boldsymbol{u}_d^*$.

In general, the default solution is the safest or one of the safest options for satisfying the constraints related to risk mitigation. Which form it takes is of course very problem-dependent. Note the following.

[2]A set $\mathcal{X}$ is said to be convex, if for all $x_1, x_2 \in \mathcal{X}$ and $\alpha \in [0, 1]$, then the point $\alpha x_1 + (1-\alpha)x_2$ also belongs to $\mathcal{X}$.

[3]A function $f : \mathcal{X} \to \mathbb{R}$ is a convex function, if for all $x_1, x_2 \in \mathcal{X}$ and $\alpha \in [0, 1]$, then $f(\alpha x_1 + (1-\alpha)x_2) \le \alpha f(x_1) + (1-\alpha)f(x_2)$.

1) The default solution does not have to be a solution to an optimization problem, but if a robustly feasible solution is available, then that can serve as a default solution.
2) It can possibly belong to a set, say $\mathcal{U}'$, which is larger than $\mathcal{U}$.

A stochastic MPC is obtained by solving Problem (13) in a receding horizon fashion. However, Problem (13) is difficult to solve, because a probabilistic constraint is in general nonconvex with respect to the optimization variables. The randomized strategies in [9]–[17] provide computationally tractable approximate solutions to CCPs, which are discussed in Section II-D.

### D. Scenario-Based Approach to Solve CCPs

To find an approximate solution to a CCP, we employ the scenario-based randomized approach described in [9]–[12], [17], and [18]. Here, we illustrate the idea of the approach using a CCP obtained by only considering the first CC of Problem (13). From now on, we will not explicitly state the state dynamics (4) and the control law (7) in the problems formulations. The CCP with the first CC of Problem (13) is

$$\begin{aligned}&\min_{\Gamma_n,\Theta_n}\ J(\Gamma_n,\Theta_n),\\ &\text{s.t. } \mathbb{P}\{\mathbf{w}_n\in\mathbb{W}:\mathbf{u}_n(\mathbf{w}_n)\in\mathcal{U}\cap f(\mathbf{u}_n,\mathbf{w}_n)\le 0\}\ge 1-\epsilon. \end{aligned}\tag{14}$$

To find a scenario-based approximate solution of Problem (14), we generate $N_r$ independent realizations of the disturbance $\mathbf{w}_n$ according to the given probability distribution and replace Problem (14) with

$$\begin{aligned}&\min_{\Gamma_n,\Theta_n}\quad J(\Gamma_n,\Theta_n)\\ &\text{s.t. } \mathbf{u}_n\left(\mathbf{w}_n^{(k)}\right)\in\mathcal{U},\quad f\left(\mathbf{u}_n\left(\mathbf{w}_n^{(k)}\right),\mathbf{w}_n^{(k)}\right)\le 0\\ &\qquad \text{for } k=1,2,\dots,N_r \end{aligned}\tag{15}$$

that is, we replace the probabilistic constraint in Problem (14) with $N_r$ constraints, each corresponding to an independent realization of the disturbance vector $\mathbf{w}_n$. Problem (15) is called a *scenario problem*, and it can be solved by resorting to standard convex optimization techniques.[4] Moreover, according to the scenario theorem stated in the following, the scenario solution provides with high confidence a feasible solution to the chance-constrained Problem (14), provided $N_r$ is chosen large enough.

*Theorem 1 [32]:* If the number of scenarios $N_r$ used in a scenario problem satisfies

$$\sum_{i=0}^{d-1}\binom{N_r}{i}\epsilon^i(1-\epsilon)^{N_r-i}\le\beta \tag{16}$$

where $\epsilon\in(0,1)$, $\beta\in(0,1)$, and $d$ is the number of optimization variables, then the scenario solution is feasible for the original CCP with confidence $1-\beta$.

Theorem 1 holds true under the assumption that the scenario problem is always feasible and has a unique solution.[5] Also, note that the number of optimization variables, $d$, can be replaced by Helly's dimension or an upper bound on the number of support constraints (see [33] for details). Bounds on the number of scenarios, $N_r$, for particular cases in an MPC context are derived in [34]–[37].

Let $\Gamma_n^*$ and $\Theta_n^*$ denote the solution to the scenario Problem (15). We will also refer to the policy, $\boldsymbol{u}_n^*=\Theta_n^*\mathbf{w}_n+\Gamma_n^*$, as the solution to the scenario problem in this paper. Note that $\Gamma_n^*$ and $\Theta_n^*$ are stochastic, because they depend on the $N_r$ drawn scenarios of $\mathbf{w}_n$ in Problem (15). Theorem 1 says that if the number of scenarios $N_r$ used in Problem (15) satisfies (16), then the following holds true:

$$\mathbb{P}^{N_r}\left\{\mathbb{P}\left\{\boldsymbol{u}_n^*\in\mathcal{U}\cap f\left(\boldsymbol{u}_n^*,\mathbf{w}_n\right)\le 0\right\}\ge 1-\epsilon\right\}\ge 1-\beta$$

where $\mathbb{P}^{N_r}$ is the probability measure on the $N_r$ extracted samples of $\mathbf{w}_n$ and $\beta$ is a confidence parameter.

The parameter $\beta$ can be explained as follows [10], [12]. We cannot guarantee that the scenario solution is always feasible for Problem (14), because it might happen that the $N_r$ extracted realizations are not representative enough. However, if we meet the criterion in (16), then the probability of such an event is less than $\beta$. In [22] and [33], explicit expressions to compute $N_r$ are described, e.g., it is shown in [22] that (16) holds true whenever $N_r$ satisfies

$$N_r\ge\frac{d-1+\ln(1/\beta)+\sqrt{2(d-1)\ln(1/\beta)}}{\epsilon}. \tag{17}$$

$N_r$ depends logarithmically on $\beta$, and hence $\beta$ can be chosen very small (e.g., $10^{-6}$) without increasing $N_r$ much. The scenario-based optimization problems can be solved by standard convex optimization solvers as, e.g., used by YALMIP [38] and CVX [39].

## III. Some Approaches to Find Approximate Solutions to an M-CCP

In this section, we first discuss some existing approaches to find an approximate solution to Problem (13), before we present the intuitive idea behind the *OTI algorithm* to be introduced in Section IV.

### A. Some Possible Ways to Solve Problem (13)

1) There are some cases when a CCP or an M-CCP is convex with respect to the optimization variables, e.g., when $\mathbf{w}_n$ is normally distributed with linear inequality constraints or when the distribution of $\mathbf{w}_n$ is log-concave [40], [41].
2) A naive application of the scenario approach to the multiple chance-constrained Problem (13) is to set $\epsilon=\epsilon_V$. However, the small value of $\epsilon_V\ll\epsilon$ would lead to a very large number of constraints in the scenario problem

[4] Strictly speaking, not all convex optimization problems are computationally tractable. However, it is a fact that important classes of convex problems are tractable in practice, possibly after reformulation. Sufficient conditions for computational tractability are known and are related to the existence of suitable (self-concordant) barrier functions for interior point methods or to the oracle complexity in black-box methods [29]–[31].

[5] When the scenario problem is feasible, the existence of a unique solution can be easily enforced by a tie-break rule and possibly by constraining the domain so as to avoid that the optimal solution drifts toward infinity [32]. In general, replacing $d-1$ with $d$ in (16) is enough to guarantee that the probability that a scenario solution is found which does not satisfy the chance-constrained optimization is no larger than $\beta$ (see [17] for details.)

[see (16)]. The computational effort is, therefore, large, and thus this is not a viable option in general.

3) In [6] and [19], scenario problems with multiple CCs are considered and it is suggested to apply the scenario approach individually to each CC. Moreover, they provide ways to reduce the computational burden of the corresponding scenario problem. Those results improve considerably on the naive approach when the CCs are sufficiently decoupled, i.e., when each CC involves different decision variables. This decoupling, which is typical in some important applications, e.g., in production planning and in portfolio optimization [6], is not present for applications where the constraints involve the same output variables and decision variables, which is the type of constraints that are motivating the proposed algorithm in this paper.
4) One approach is to robustify the constraints such that they must be satisfied for all possible realizations of the disturbances and solve a robust MPC problem. This may be difficult if the disturbances have a very large support, in which case a combination of robust MPC and a scenario optimization problem can be used to determine a range or set to which the disturbances belong most of the time [35], [36].

### *B. Intuitive Description of the OTI Algorithm*

Here, we give a description of the noniterative OTI algorithm, which is an improved version of the iterative algorithm in [42]. It has three main steps as follows.

*1) Optimization:* We first solve an optimization problem with only the first CC, i.e., we solve Problem (14). We generate $N_r$ independent realizations of the disturbance $\mathbf{w}_n$, according to the given probability distribution, and replace Problem (14) with Problem (15). We then solve Problem (15) and find a scenario solution, say $\boldsymbol{u}_n^*$.

*2) Testing:* In the second step, we test the solution $\boldsymbol{u}_n^*$ against the second CC by resorting to a Monte Carlo sample of $N_T$ new scenarios. This is computationally cheap, since no optimization is performed. If the number of scenarios violating the constraint is below a selected threshold, we use the solution $\boldsymbol{u}_n^*$ and provide a certificate with a probabilistic guarantee that the solution satisfies both CCs. Otherwise, we improve the obtained solution.

*3) Improving:* If the solution $\boldsymbol{u}_n^*$ fails the aforementioned test, we test the default solution $\boldsymbol{u}_d^*$ (see Section II-C) against the second CC using the same $N_T$ scenarios. If the violations are less than the selected threshold, we solve a 1-D scenario problem by moving $\boldsymbol{u}_n^*$ in the direction of the default solution $\boldsymbol{u}_d^*$ along the line $(1-\alpha)\boldsymbol{u}_n^* + \alpha\boldsymbol{u}_d^*$, where $\alpha \in (0, 1]$. In the scenario problem, we minimize the value of $\alpha$ and consider the scenarios $\mathbf{w}_n \in \mathbb{W}$ that violate the risk-mitigation (second) constraint with $\boldsymbol{u}_n^*$ to ensure improvement but satisfy the constraint with $\boldsymbol{u}_d^*$ to ensure the feasibility of the scenario problem. However, if the default solution fails the test against the second CC, then we exit the algorithm and inform system operators that no solution can be found. In the case of control of rivers, flood operations can be pursued on such occasions.

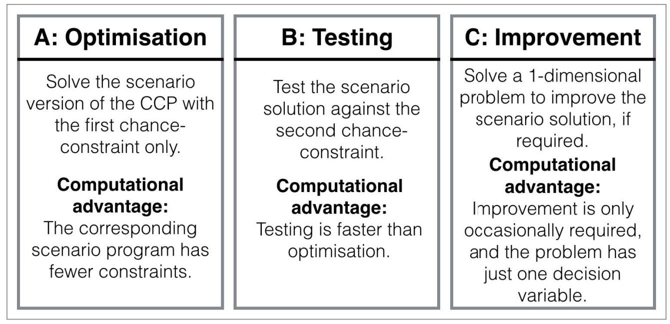


Fig. 1. *OTI algorithm* (basic idea).

The sample sizes $N_r$ and $N_T$ are computed in such a way that the solution comes with precise probabilistic guarantees. In particular, if the algorithm successfully ends after the testing procedure, a certificate is delivered that states that both the CCs in Problem (13) are satisfied with high confidence. Otherwise, the first CC of Problem (13) can be tested with the improved solution, and depending on the outcome of the test, a certificate with the satisfaction of both CCs or a certificate guaranteeing only the second CC is delivered. Fig. 1 summarizes the idea of the algorithm with an emphasis on the computational advantages it provides.

## IV. Optimization, Testing, and Improving Algorithm to Solve Problem (13)

In this section, we formally state the OTI algorithm, followed by theoretical justifications of sample sizes and allowed violation probabilities. For notational simplicity, we omit the subscripts from the $\mathbf{u}_n$, $\boldsymbol{u}_n^*$, and $\mathbf{w}_n$ vectors.

### *A. OTI Algorithm*

The following parameters must be chosen by the user.

1) $\beta = \delta/7$, where $\delta \in (0, 1)$ is the overall probability of failure (POF) of the algorithm. It is the probability that the algorithm delivers a wrong certificate. $\delta$ should be set very low, e.g., $10^{-7}$ (see also Remark 1).
2) $\epsilon$ and $\epsilon_V$ give the allowed violation probabilities of the two CCs in Problem (13).
3) $\varrho > 0$ and $\varrho_V > 0$ are the margins in the tests. The tests are devised such that the estimated violation probability $\hat{p}$ must be less than $\epsilon - \varrho$ in order to pass the test, where $\epsilon$ is the allowed violation probability. $\varrho$ and $\varrho_V$ can, e.g., be chosen as $\varrho = \frac{\epsilon}{2}$ and $\varrho_V = (\frac{\epsilon_V}{2})$.

Next we present Algorithm 1. The technical explanation for the sample sizes $N_T$ and $N_I$ in Steps B-1 and C-4 is given in Section IV-B1, and the allowed violation probability $\epsilon_\alpha$ in Step C-1 is explained in Sections IV-B2 and IV-B3.

*Algorithm 1:*

**A-Optimization** (Find a scenario solution $\boldsymbol{u}^*$ of Problem (14) using Problem (15)):

A-1) Compute the smallest number (of scenarios) $N_r$ that satisfies (16), given the violation probability $\epsilon$ for the first CC, the number of decision variables $d$, and the confidence parameter $\beta$.

A-2) Find the solution, $\boldsymbol{u}^*$, of Problem (15).[6]

**B-Testing** (Test the solution, $\boldsymbol{u}^*$, against the second CC in Problem (13)):

B-1) Compute the smallest number (of scenarios) $N_T$ that satisfies $\sum_{i=0}^{\lfloor N_T(\epsilon_V-\varrho_V)\rfloor}\binom{N_T}{i}\epsilon_V^i(1-\epsilon_V)^{N_T-i} < \beta$, given the violation probability $\epsilon_V$ for the second CC, and the margin $\varrho_V$.

B-2) Generate $N_T$ scenarios of $\mathbf{w} \in \mathbb{W}$, and compute the violation probability estimate for the second CC:

$$\hat{p} = \frac{1}{N_T}\sum_{k=1}^{N_T}\mathbb{1}\{g(\boldsymbol{u}^*, \mathbf{w}^{(k)}) > 0\}, \tag{18}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

B-3) If $\hat{p} \leq \epsilon_V - \varrho_V$, then the test is passed. Exit the algorithm and forward the following solution,

$$\begin{cases} \boldsymbol{u}^* \\ \text{with a certificate:} \\ \text{“}\mathbb{P}\{\mathbf{w}\in\mathbb{W} : \boldsymbol{u}^*\in\mathcal{U}\ \cap\ f(\boldsymbol{u}^*,\mathbf{w})\leq 0\}\geq 1-\epsilon, \\ \ \mathbb{P}\{\mathbf{w}\in\mathbb{W} : g(\boldsymbol{u}^*,\mathbf{w})\leq 0\}\geq 1-\epsilon_V.\text{”} \end{cases}$$

Otherwise, test the default solution $\boldsymbol{u}_d^* \in \mathcal{U}' \supseteq \mathcal{U}$ (see Section II-C) by computing the violation probability estimate,

$$\hat{p}_d = \frac{1}{N_T}\sum_{k=1}^{N_T}\mathbb{1}\{g(\boldsymbol{u}_d^*, \mathbf{w}^{(k)}) > 0\}. \tag{19}$$

If $\hat{p}_d \leq \epsilon_V - \varrho_V$, then proceed to the next step. Otherwise, exit the algorithm with no solution found.

**Preparation for the Improving step**

P-1) Save, in a set $\mathcal{Q}$, the $\mathbf{w}$-scenarios which violate the constraint '$g(\mathbf{u},\mathbf{w}) \leq 0$' with $\boldsymbol{u}^*$, but satisfy the constraint with $\boldsymbol{u}_d^*$, i.e., $\mathcal{Q} = \{\mathbf{w}^{(k)} | g(\boldsymbol{u}^*, \mathbf{w}^{(k)}) > 0 \wedge g(\boldsymbol{u}_d^*, \mathbf{w}^{(k)}) \leq 0,\ \text{for } k = 1, 2, \ldots, N_T\}$.

**C-Improving** (Improve the solution $\boldsymbol{u}^*$ in the direction of $\boldsymbol{u}_d^*$):

C-1) Compute $\epsilon_\alpha = (\epsilon_V - \hat{\bar{\epsilon}}_d)/\hat{\bar{\epsilon}}_T$, where $\hat{\bar{\epsilon}}_d$ and $\hat{\bar{\epsilon}}_T$ are obtained by solving: $\sum_{i=0}^{\hat{p}_d N_T}\binom{N_T}{i}\hat{\bar{\epsilon}}_d^i(1-\hat{\bar{\epsilon}}_d)^{N_T-i} = \beta$ and $\sum_{i=0}^{\hat{p}_T N_T}\binom{N_T}{i}\hat{\bar{\epsilon}}_T^i(1-\hat{\bar{\epsilon}}_T)^{N_T-i} = \beta$ respectively, $\hat{p}_T = |\mathcal{Q}|/N_T$, and $|\mathcal{Q}|$ is the cardinality of the set $\mathcal{Q}$ in Step P-1.

$\epsilon_\alpha$ is the allowed conditional[7] violation probability, for the CCP corresponding to the one-dimensional scenario Problem (20) given below, and compute $N_\alpha$ (the number of scenarios required) according to $N_\alpha = \ln\beta/\ln(1-\epsilon_\alpha)$.

C-2) If $N_\alpha > |\mathcal{Q}|$, then generate $N_\alpha - |\mathcal{Q}|$ realisations of $\mathbf{w} \in \mathbb{W}$, such that the constraint $g(\boldsymbol{u}^*, \mathbf{w}) \leq 0$ is violated, but $g(\boldsymbol{u}_d^*, \mathbf{w}) \leq 0$ is satisfied. Include the generated realisations in $\mathcal{Q}$.

[6]There is a possibility that the scenario Problem (15) is infeasible with the drawn scenarios. In such circumstances, the optimization schemes introduced in [33] can be used. However, for the schemes in [33], one would need to recalculate the overall POF of the algorithm. For a discussion on the theory of the scenario approach for infeasible problems (see also [8]).

[7]$\epsilon_\alpha$ is the allowed violation probability, conditioned on the set: $\{\mathbf{w} \in \mathbb{W} \mid g(\boldsymbol{u}^*, \mathbf{w}) > 0 \wedge g(\boldsymbol{u}_d^*, \mathbf{w}) \leq 0\}$.

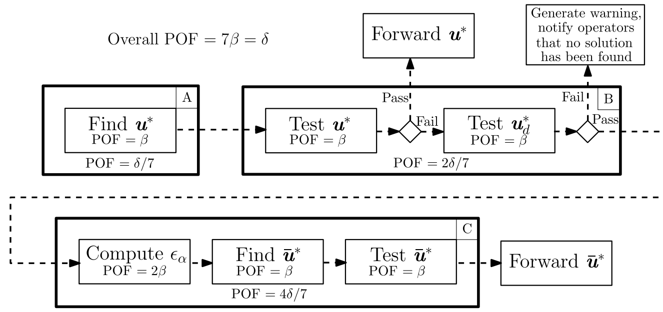


Fig. 2. *OTI algorithm.*

C-3) Let $(\Gamma_\alpha, \Theta_\alpha) = (1-\alpha)(\Gamma^*, \Theta^*) + \alpha(\Gamma_d^*, \Theta_d^*)$, where $(\Gamma^*, \Theta^*)$ is the parametrisation of the solution, $\boldsymbol{u}^*$, obtained in Step A and $(\Gamma_d^*, \Theta_d^*)$ is the parametrisation of the default solution, $\boldsymbol{u}_d^*$, and $\alpha \in (0, 1]$. Denote the control policy corresponding to $(\Gamma_\alpha, \Theta_\alpha)$ by $\bar{\mathbf{u}}_\alpha(\cdot)$ Solve the following one-dimensional scenario problem,

$$\begin{aligned} &\min_{\alpha\in(0,1]}\ J(\Gamma_\alpha, \Theta_\alpha) \\ &\text{s.t. } \bar{\mathbf{u}}_\alpha(\mathbf{w}^{(k)}) \in \mathcal{U}',\quad g(\bar{\mathbf{u}}_\alpha, \mathbf{w}^{(k)}) \leq 0 \\ &\quad\ \text{for } k = 1, 2, \ldots, N_\alpha, \end{aligned} \tag{20}$$

The scenarios $\mathbf{w}^{(k)}$ are drawn (without replacement) from the set $\mathcal{Q}$. Denote the solution by $\alpha^*$ and the corresponding control policy by $\bar{\boldsymbol{u}}^*$.

**Test against the first CC:**

C-4) Compute the smallest number (of scenarios) $N_I$ that satisfies $\sum_{i=0}^{\lfloor N_I(\epsilon-\varrho)\rfloor}\binom{N_I}{i}\epsilon^i(1-\epsilon)^{N_I-i} < \beta$, given the violation probability $\epsilon$ for the first CC, and the margin $\varrho$.

C-5) Generate $N_I$ scenarios of $\mathbf{w} \in \mathbb{W}$, and compute the violation probability estimate for the first CC:

$$\hat{p}_I = \frac{1}{N_I}\sum_{k=1}^{N_I}\mathbb{1}\{f(\bar{\boldsymbol{u}}^*, \mathbf{w}^{(k)}) > 0\}. \tag{21}$$

C-6) If $\hat{p}_I \leq \epsilon - \varrho$, then the test is passed. Forward the following solution,

$$\begin{cases} \bar{\boldsymbol{u}}^* \\ \text{with a certificate:} \\ \text{“}\mathbb{P}\{\mathbf{w}\in\mathbb{W} : \bar{\boldsymbol{u}}^*\in\mathcal{U}'\ \cap\ f(\bar{\boldsymbol{u}}^*,\mathbf{w})\leq 0\}\geq 1-\epsilon, \\ \ \mathbb{P}\{\mathbf{w}\in\mathbb{W} : g(\bar{\boldsymbol{u}}^*,\mathbf{w})\leq 0\}\geq 1-\epsilon_V.\text{”} \end{cases}$$

Otherwise, forward the following solution,

$$\begin{cases} \bar{\boldsymbol{u}}^* \\ \text{with a certificate:} \\ \text{“}\mathbb{P}\{\mathbf{w}\in\mathbb{W} : \bar{\boldsymbol{u}}^*\in\mathcal{U}'\ \cap\ g(\bar{\boldsymbol{u}}^*,\mathbf{w})\leq 0\}\geq 1-\epsilon_V.\text{”} \end{cases}$$

Fig. 2 summarizes the steps of the algorithm and their POFs.

*Remarks:*

1) As there are seven $\beta$ terms in the algorithm, the POF is $\delta$ when each $\beta$ term is $\delta/7$. There is one term in the optimization phase associated with the scenario problem and two terms in the testing phase associated

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



with the two tests in Step B-3. There are also two terms associated with the computations of the upper bounds required for computing $\epsilon_\alpha$ in the first step of the improving phase (see Sections IV-B2 and IV-B3) and finally two terms in the improving phase related to the scenario problem in Step C-3 and the test in Step C-6. The $\beta$ terms are added to bound the probability of the event where any of the randomized steps in the algorithm fail. The validity of adding $\beta$ terms is explained in Section IV-B5.

2) If the algorithm exits in Step B-3 with no solution, then the control can be shifted to some alternative operations. For example, for river control problems, this could mean changing the operational mode to flood operations.
3) If the number of scenarios $N_\alpha$, computed in Step C-1, is larger than what the computational resources can deal with, we can replace $N_\alpha$ with an upper limit $N_{\max}$. In that case, the found solution must be tested against both CCs. If the tests are passed, the probability guarantees remain valid. However, this scheme requires one extra $\beta$ term (corresponding to the test against the second CC) in the overall POF.
4) The proposed algorithm is developed specifically for cases where $\epsilon_V \ll \epsilon$, and in such cases, the algorithm offers large computational savings.
5) A sampling-and-discarding approach [11] can be introduced to deal with possible conservatism in ensuring feasibility of the scenario problem constraints. In this approach, some scenarios are removed in order to further decrease the value of the cost function, with an increase in the violation probability of the constraints up to an allowed limit (for details see [11], [17], and [43]).
6) A possible extension of the algorithm to problems with more than two CCs is described in [28].

*Discussion:*

Other works that exploit the fact that testing is computationally cheaper than optimization include [21]–[24]. Different from those works, the found solution in the optimization step is here tested against a different, but related CC.

Note that an approach based on a combination of robust MPC and scenario optimization [35], [36] can also be used to find a solution in the optimization step, which is feasible with confidence $1-\beta$.

The number of scenarios required in the optimization step of a stochastic MPC problem can be bounded in different ways by considering the problem formulation and the system structure, and it is problem-dependent, which bounds lead to the smallest scenarios [34]–[37].

In the problem formulation [see Problem (13)], we have used a so-called joint CC. Stagewise CCs of the form $\mathbb{P}\{g(x_k,u_k)\le 0\}\ge 1-\epsilon_k$, for $k=n+1,\ldots,n+M$, are also commonly used [34], [35], [37]. Considering stagewise CCs in the formulation can be favorable in terms of the sample complexity. However, adapting our scheme to the stagewise setup will require several modifications in the problem formulation, and that is not considered due to space limitations.

### B. Technical Explanation

In this section, we explain how the required number of scenarios for the tests can be found and how $\epsilon_\alpha$ in Step C-1 is computed.

*1) Number of Scenarios Required for Testing:* The tests carried out in the algorithm consist of drawing a number $N$ of independent realizations of $\mathbf{w}\in\mathbb{W}$ and calculate the empirical frequency of constraint violation. Consider a constraint of the type $\mathbb{P}\{\mathbf{w}\in\mathbb{W}: h(\boldsymbol{u}^*,\mathbf{w})\le 0\}\ge 1-\epsilon$, and let $p$ be the actual probability that the constraint is violated, i.e., $p=\mathbb{P}\{\mathbf{w}\in\mathbb{W}: h(\boldsymbol{u}^*,\mathbf{w})>0\}$. We evaluate the constraint $h(\boldsymbol{u}^*,\mathbf{w}^{(k)})\le 0$ for the independently drawn realizations $\mathbf{w}^{(k)}\in\mathbb{W}$, $k=1,2,\ldots,N$. The corresponding Bernoulli random variables $B_k$ take the value 1 if the constraint is violated, and 0 otherwise, i.e., $B_k=\mathbb{1}\{h(\boldsymbol{u}^*,\mathbf{w}^{(k)})>0\}$. The probability $\mathbb{P}(B_k=1)=\mathbb{P}\{\mathbf{w}^{(k)}\in\mathbb{W}: h(\boldsymbol{u}^*,\mathbf{w}^{(k)})>0\}=p$.

An estimate of the violation probability can, therefore, be obtained as

$$\hat{p}=\frac{1}{N}\sum_{k=1}^{N}B_k=\frac{1}{N}\sum_{k=1}^{N}\mathbb{1}\{h(\boldsymbol{u}^*,\mathbf{w}^{(k)})>0\}. \tag{22}$$

The test is passed if $\hat{p}\le\epsilon-\varrho$, where $\varrho\in(0,\epsilon]$ is a user chosen margin.

The number of realizations $N$ should be chosen large enough, so that the probability of passing the test when the actual violation probability $p$ is larger than the allowed one $\epsilon$ is less than $\beta$, i.e.,

$$\mathbb{P}(\hat{p}\le\epsilon-\varrho\,|\,p>\epsilon)<\beta. \tag{23}$$

In order to find $N$, we observe that the total number of violations, $V$, in $N$ Bernoulli trials is a Binomial random variable, and the probability that $V=\upsilon$ is

$$\mathbb{P}(V=\upsilon,p)=\binom{N}{\upsilon}p^{\upsilon}(1-p)^{N-\upsilon}. \tag{24}$$

For a given $p=\mu$, it follows that:

$$\mathbb{P}(\hat{p}\le\epsilon-\varrho\,|\,p=\mu)=\sum_{i=0}^{\lfloor N(\epsilon-\varrho)\rfloor}\binom{N}{i}\mu^{i}(1-\mu)^{N-i} \tag{25}$$

where $\lfloor N(\epsilon-\varrho)\rfloor$ is the largest integer smaller or equal to $N(\epsilon-\varrho)$. Furthermore,

$$\begin{aligned}\mathbb{P}(\hat{p}\le\epsilon-\varrho\,|\,p=\mu>\epsilon)&\le\sup_{\mu>\epsilon}\sum_{i=0}^{\lfloor N(\epsilon-\varrho)\rfloor}\binom{N}{i}\mu^{i}(1-\mu)^{N-i},\\&=\sum_{i=0}^{\lfloor N(\epsilon-\varrho)\rfloor}\binom{N}{i}\epsilon^{i}(1-\epsilon)^{N-i}.\end{aligned}$$

The last equality holds because the expression before the equality sign is monotonically decreasing in $\mu$. From the above considerations, we have the following result.

*Lemma 1:* The minimum number of realizations, $N$, required to guarantee $\mathbb{P}(\hat{p}\le\epsilon-\varrho\,|\,p>\epsilon)<\beta$, where $\epsilon$, $p$, and $\hat{p}$ are the allowed, actual, and empirical violation probabilities, is given by

$$\min\{N\in\mathbb{N}:\sum_{i=0}^{\lfloor N(\epsilon-\varrho)\rfloor}\binom{N}{i}\epsilon^{i}(1-\epsilon)^{N-i}<\beta\} \tag{26}$$

where $\mathbb{N}$ is the set of natural numbers.

From Lemma 1, it follows that $N_T$ in Step B-1 of the algorithm can be found from:

$$\min\left\{N_T \in \mathbb{N} : \sum_{i=0}^{\lfloor N_T(\epsilon_V-\varrho_V)\rfloor} \binom{N_T}{i}\epsilon_V^i(1-\epsilon_V)^{N_T-i} < \beta\right\} \tag{27}$$

and $N_I$ in Step C-4 can be found from

$$\min\left\{N_I \in \mathbb{N} : \sum_{i=0}^{\lfloor N_I(\epsilon-\varrho)\rfloor} \binom{N_I}{i}\epsilon^i(1-\epsilon)^{N_I-i} < \beta\right\}. \tag{28}$$

*2) Violation Probability for the 1-D Scenario Problem:* In the improvement step (Step C-3), we solve a 1-D scenario Problem (20), where the scenarios are drawn from the set

$$\mathcal{T} = \left\{\mathbf{w} \in \mathbb{W} \mid g(\boldsymbol{u}^*, \mathbf{w}) > 0 \wedge g\left(\boldsymbol{u}_d^*, \mathbf{w}\right) \le 0\right\} \subseteq \mathbb{W}. \tag{29}$$

The set $\mathcal{T}$ contains the realizations of $\mathbf{w}$ that satisfy the second CC with $\boldsymbol{u}_d^*$ [this grants feasibility of Problem (20)] but does not satisfy the CC with $\boldsymbol{u}^*$. Problem (20) finds a policy along the line from $\boldsymbol{u}^*$ to $\boldsymbol{u}_d^*$, where the line is formulated in the $(\Gamma, \Theta)$ space. In general, selecting a policy which is a convex combination of two policies does not guarantee that the violation probability of the obtained policy is small, although it is bounded by the sum of the violation probabilities of the original two policies due to the convexity assumption. However, in the OTI algorithm, the violation probability of the 1-D optimization problem is chosen such that one can guarantee with confidence $1-\beta$ that the violation probability for the second CC is no more than $\epsilon_V$ for the found control policy. This is shown in Theorem 2.

In addition to the set $\mathcal{T}$, we also need the set $\mathrm{VS}(\boldsymbol{u}_d^*) = \{\mathbf{w} \in \mathbb{W} \mid g(\boldsymbol{u}_d^*, \mathbf{w}) > 0\}$ in Theorem 2.

*Theorem 2:* Assume that $\mathbb{P}(\mathcal{T}) \le \hat{\hat{\epsilon}}_T$ and $\mathbb{P}(\mathrm{VS}(\boldsymbol{u}_d^*)) \le \hat{\hat{\epsilon}}_d$. Let

$$\epsilon_\alpha \le \frac{\epsilon_V - \hat{\hat{\epsilon}}_d}{\hat{\hat{\epsilon}}_T} \tag{30}$$

and furthermore let the number of scenarios for the scenario Problem (20) be $N_\alpha = \ln\beta/\ln(1-\epsilon_\alpha)$, where the scenarios are independently sampled according to the probability distribution of $\mathbf{w} \in \mathbb{W}$ conditioned on that $\mathbf{w}$ belongs to $\mathcal{T}$. Then, the control policy $\bar{\boldsymbol{u}}^* = (1-\alpha^*)\boldsymbol{u}^*+\alpha^*\boldsymbol{u}_d^*$, corresponding to the solution of Problem (20), violates the constraint "$g(\bar{\boldsymbol{u}}^*, \mathbf{w}) \le 0$" with a probability no more than $\epsilon_V$ with a confidence $1-\beta$.

A proof of Theorem 2 is given in the Appendix. To compute the allowed violation probability $\epsilon_\alpha$ in (30), we need the upper bounds $\hat{\hat{\epsilon}}_d$ and $\hat{\hat{\epsilon}}_T$, which are discussed in what follows.

*3) Upper Bounds on $\mathbb{P}(\mathcal{T})$ and $\mathbb{P}(VS(\boldsymbol{u}_d^*))$:*[8] For a given solution, the number of violations of a CC in the $N$ Bernoulli trials is a binomial random variable $V$. The probability, $\mathbb{P}(V = \upsilon, p)$, is given by (24), and the probability that the number of violations is $\upsilon$ or smaller is given by

$$\mathbb{F}\left(\frac{\upsilon}{N}, p\right) = \mathbb{P}\left(\hat{p} \le \frac{\upsilon}{N}, p\right) = \sum_{i=0}^{\upsilon}\mathbb{P}(V = i, p). \tag{31}$$

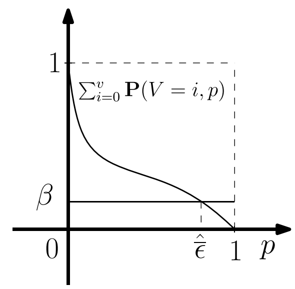


Fig. 3. $\mathbb{F}((\upsilon/N_T), p)$ as a function of $p$.

The function $\mathbb{F}((\upsilon/N), p)$ is nonincreasing with respect to $p$. When $\upsilon$ violations have occurred, the desired upper bound is obtained by finding the largest value of $p$, such that $\mathbb{F}((\upsilon/N), p) \ge \beta$. This is shown in Fig. 3, where $\mathbb{F}((\upsilon/N), p)$ is plotted as a function of $p$ for a given $\upsilon$, and the upper bound $\hat{\hat{\epsilon}}$ is characterized by $\mathbb{F}((\upsilon/N), \hat{\hat{\epsilon}}) = \beta$.

Therefore, the upper bounds $\hat{\hat{\epsilon}}_T$ and $\hat{\hat{\epsilon}}_d$ on $\mathbb{P}(\mathcal{T})$ and $\mathbb{P}(\mathrm{VS}(\boldsymbol{u}_d^*))$ can be found by solving the following equations:

$$\sum_{i=0}^{\hat{p}_T N_T} \binom{N_T}{i}\hat{\hat{\epsilon}}_T^i(1-\hat{\hat{\epsilon}}_T)^{N_T-i} = \beta \tag{32}$$

$$\sum_{i=0}^{\hat{p}_d N_T} \binom{N_T}{i}\hat{\hat{\epsilon}}_d^i(1-\hat{\hat{\epsilon}}_d)^{N_T-i} = \beta \tag{33}$$

where $\hat{p}_d$ is given by (19) and $\hat{p}_T = |\mathcal{Q}|/N_T$ where $|\mathcal{Q}|$ is the cardinality of the set $\mathcal{Q}$ generated in preparation for the improving step (see Steps P-1, C-1, and C-2 of the algorithm).

*4) Strict Positivity of $\epsilon_\alpha$ in (30):* The above procedure is guaranteed to produce an upper bound on $\mathbb{P}(\mathrm{VS}(\boldsymbol{u}_d^*))$ which is less than $\epsilon_V$, which implies that $\epsilon_\alpha > 0$. This can be seen as follows. From Step B-1 and (26), it follows that $N_T$ is chosen such that:

$$\sum_{i=0}^{\lfloor N_T(\epsilon_V-\varrho_V)\rfloor} \binom{N_T}{i}\epsilon_V^i(1-\epsilon_V)^{N_T-i} < \beta \tag{34}$$

holds true. Furthermore, the default solution must have passed the test in Step B-3, and hence $\hat{p}_d \le \epsilon_V - \varrho_V$ [see (19)]. In view of the monotonicity of (31) with respect to $p$, by comparing (33) and (34), we conclude that $\hat{\hat{\epsilon}}_d < \epsilon_V$.

*5) Validity of Adding $\beta$ Terms in the Algorithm:* The $\beta$ terms are added to bound the probability of the event where any of the randomized steps in the OTI algorithm fails. This is valid because of the following argument, where, for simplicity, we consider just the optimization step followed by the testing of $\boldsymbol{u}^*$ with respect to the second CC. The reasoning goes through similarly for all the other POFs. We want to bound the probability of the event: $\{\boldsymbol{u}^*$ violates the first CC$\}$ $\cup$ $\{$issue a false claim about $\boldsymbol{u}^*$ based on the testing outcome$\}$. This is an event over $N_r + N_T$ samples and whose probability can be bounded by $\mathbb{P}^{N_r+N_T}\{\boldsymbol{u}^*$ violates the first CC$\}$ $+$

[8]The idea behind finding the upper bounds on the violation probabilities is the same as used in the Clopper–Pearson method to find a confidence interval for the parameter of the binomial distribution function (for details see [44] and the references therein).

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



$\mathbb{P}^{N_r+N_T}\{\hat{p} \leq \epsilon_V - \rho$ **and** $\mathbb{P}\{\text{VS}(\boldsymbol{u}^*)\} > \epsilon_V\}$, where $\text{VS}(\boldsymbol{u}^*)$ is the set of all realizations $\mathbf{w} \in \mathbb{W}$ that violate the constraint "$g(\boldsymbol{u}^*, \mathbf{w}) \leq 0$." Since $\boldsymbol{u}^*$ depends only on the first $N_r$ scenarios, $\mathbb{P}^{N_r+N_T}\{\boldsymbol{u}^*$ violates the first CC$\} = \mathbb{P}^{N_r}\{\boldsymbol{u}^*$ violates the first CC$\}$, which is bounded by $\beta$ in view of Theorem 1. Now, $\mathbb{P}^{N_r+N_T}\{\hat{p} \leq \epsilon_V - \rho$ **and** $\mathbb{P}\{\text{VS}(\boldsymbol{u}^*)\} > \epsilon_V\} = \mathbb{P}^{N_T}\{\hat{p} \leq \epsilon_V - \rho \mid \mathbb{P}\{\text{VS}(\boldsymbol{u}^*)\} > \epsilon_V\} \times \mathbb{P}^{N_r}\{\mathbb{P}\{\text{VS}(\boldsymbol{u}^*)\} > \epsilon_V\}$, which is bounded by $\mathbb{P}^{N_T}\{\hat{p} \leq \epsilon_V - \rho \mid \mathbb{P}\{\text{VS}(\boldsymbol{u}^*)\} > \epsilon_V\}$, and this probability is also less than $\beta$ by (27) and Lemma 1.

## V. Application to a River Control Problem

In this section, we show how a river control problem can be formulated as a CCP with two CCs within a stochastic MPC setting, and we apply the proposed algorithm to historical data from the upper part of Murray River in Australia.

### *A. River Control Problem as a CCP With Two CCs*

Water is a precious resource, and in the last few decades, large efforts have gone into improving the management of water resources, including rivers. Many works have appeared on system identification and control of rivers [7], [8], [28], [45]–[52].

Rivers have long time delays, since locations where flows can be regulated are often far away from locations where the controlled variable is measured. As a result, forecasts of unregulated in- and out-flows are required for control purposes. These forecasts are uncertain and a hard constraint, e.g., on the water level in a lake, that is dependent on uncertain flow forecasts can cause infeasibility in the control optimization problem. Therefore, it is natural to formulate the river control problem as an optimization problem with probabilistic constraints.

Like several other systems, rivers have both normal operations and operations related to risk mitigation. The objectives of normal river operations include keeping water levels in reservoirs and flow releases from reservoirs within safe limits, while the change in flows and water levels should also be less than given thresholds. These constraints can be encapsulated in one CC associated with normal operations as in (10), [8]. Risk mitigation operations can include avoiding floods and damage to infrastructure (e.g., dams), and such constraints can be expressed as the second CC associated with risk mitigation, as in (11). The aforementioned two CCs together with the objective function, $J(\Gamma_n, \Theta_n)$, in (12), form the river control problem as a CCP with two CCs [similar to Problem (13)] to be solved in a stochastic MPC setting.

The OTI algorithm proposed in Section IV can be used to find approximate solutions to the river control problems. The strategy of the algorithm, i.e., optimization, testing, and improving, is well suited to the flood risk mitigation problem, because we do not want to be overly cautious about flood risks, since most of the time there is no or a very little risk of flooding.

### *B. Application of the Proposed Algorithm to the Upper Part of Murray River*

In this section, we describe the upper part of Murray River in Australia and apply the OTI algorithm to river data.

*1) Upper Part of Murray River:* Murray River is the longest river in Australia. Fig. 4 shows a sketch of the river from Hume Reservoir to Lake Mulwala, which has a river distance of 180 km. The release from Hume is measured at Heywoods. The maximum discharge capacity from Hume is approximately 600 000 ML/day[9] at full supply level. Two unregulated rivers: Kiewa River and Ovens River join Murray River on its way to Lake Mulwala. Inflows from Kiewa and Ovens are measured at Bandiana and Peechelba, respectively.[10] There are several measuring stations upstream of Lake Mulwala on the Murray River, such as Doctors Point, Albury, Howlong, and Corowa. At the downstream end of the lake, there are three demand-based releases: at Yarrawonga Weir to the downstream part of Murray River, and to the irrigation channels: Yarrawonga Main Channel and Mulwala Canal, which have maximum flow capacities of 3170 and 10 000 ML/day, respectively. During normal operations, the water level is controlled from Hume only. The main control objectives are as follows.

1) The water level in Lake Mulwala should be kept between 124.65 and 124.9 mAHD (meter Australian Height Datum—relative to sea level), and when there is a risk of flooding, it should only cross a higher limit of 125 mAHD with a very small probability.
2) The release from Hume Reservoir should be kept between 2500 and 30 000 ML/day. However, when there is a risk of flooding, the release can be further reduced to 1000 ML/day.
3) In order to protect the river banks, the rate of decrease in the water level at Heywoods should be less than 0.20 m/day. We roughly translate this requirement, with the help of rating curves, to a constraint on the rate of decrease in the release from Hume, which should be less than 800 ML/day/day.

We use the following discrete time model of the water level in Lake Mulwala "$y_{\text{LM}}$," which is given in [48] [the sampling interval ($T_s$) is 8 h]:

$$\begin{aligned} y_{\text{LM},n+1} = {} & y_{\text{LM},n} + 10^{-6} \\ & \times [4.96 Q_{H,n-9} + 7.70 Q_{B,n-9} + 4.71 Q_{P,n-2} \\ & - 4.90 Q_{\text{DYW},n} - 7.78 Q_{\text{YMC},n} - 5.34 Q_{\text{MC},n}] \end{aligned} \tag{35}$$

where $Q_H$, $Q_B$, and $Q_P$ are the inflows from Heywoods, Bandiana, and Peechelba, and $Q_{\text{DYW}}$, $Q_{\text{YMC}}$, and $Q_{\text{MC}}$ are the releases to downstream of Yarrawonga Weir, Yarrawonga Main Channel, and Mulwala Canal. The MPC strategy uses an equivalent state space model to the model in (35) (for details see [28]).

In this paper, we assume that the flow demands from the irrigation channels, Yarrawonga Main Channel and Mulwala

[9] $1\ \text{m}^3/\text{s} = 86.4$ ML/day.

[10] Strictly speaking, flows are not measured but they are calculated from water level measurements using rating curves.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



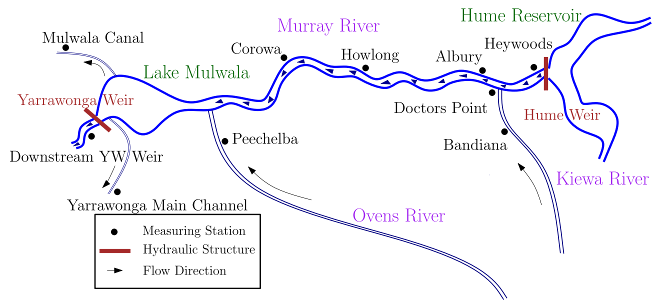


Fig. 4. Hume Reservoir to Lake Mulwala on the Murray River (plot not to scale).

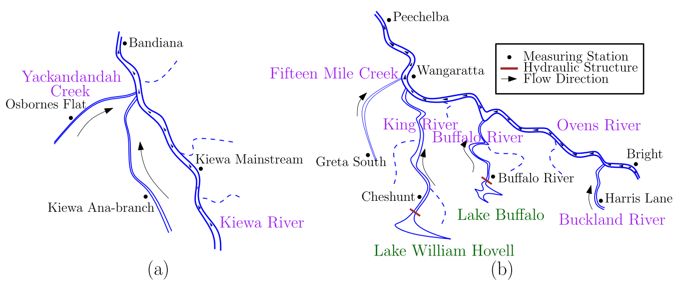


Fig. 5. (a) Kiewa and (b) Ovens Rivers (plot not to scale).

Canal, and the flow over Yarrawonga Weir are known for a finite future horizon. However, the unregulated inflows from Bandiana and Peechelba are not known. Therefore, forecast models of the flows at Bandiana and Peechelba are needed. The forecast model for the flows at Bandiana, $Q_B^f$, is [28]

$$Q^f_{B,n} = 0.99Q_{\mathrm{KMS},n-2} + 1.50Q_{\mathrm{KAB},n-2} + 1.17Q_{O,n-1} \tag{36}$$

where $Q_{\mathrm{KMS}}$, $Q_{\mathrm{KAB}}$, and $Q_O$ are the inflows at Kiewa Mainstream, Kiewa Ana-branch, and Osbornes Flat (O), which all contribute to the flows at Bandiana. A sketch of Kiewa River is given in Fig. 5(a). Similarly, the forecast model for the flows at Peechelba, $Q_P^f$, is [28]

$$\begin{aligned} Q^f_{P,n} &= 1.50Q_{Br,n-8} + 1.50Q_{\mathrm{HL},n-8} \\ &\quad + 0.67Q_{\mathrm{BR},n-8} + 1.50Q_{C,n-9} + 1.50Q_{G,n-8} \end{aligned} \tag{37}$$

where $Q_{Br}$, $Q_{\mathrm{HL}}$, $Q_{\mathrm{BR}}$, $Q_C$, and $Q_G$ are the inflows at Bright (Br), Harris Lane (HL), Buffalo River (BR), Cheshunt (C), and Greta South (G), which all contribute to the flows at Peechelba. A sketch of Ovens River is given in Fig. 5(b). More details are given in [28].

Fig. 6 shows the simulations of the flows at Bandiana and Peechelba on a validation data set, using the forecast models in (36) and (37). The blue curves show the actual recorded data and the red dashed curves show the simulation results. The models pick the trends in the data well; however, they show some discrepancies, especially at peaks. To account for the discrepancies, we introduce additive error terms $\omega_B$ and $\omega_P$ in the models (36) and (37), where $\omega_B$ and $\omega_P$ are independent and identically distributed Gaussian random variables with zero means and 50 and 200 ML/day standard deviations (s.d.), respectively. The s.d. is chosen based on the

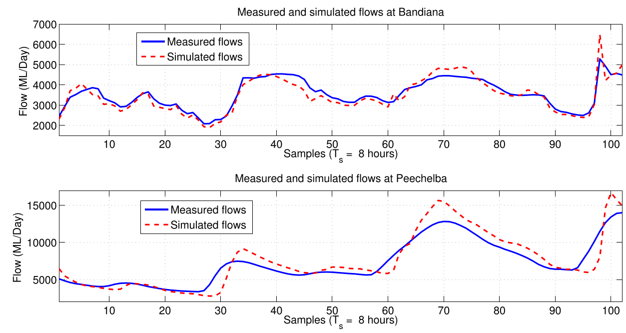


Fig. 6. Measured and simulated flows at Bandiana and Peechelba.

simulation results of the forecast models on different validation data. The choice of $\omega_B$ and $\omega_P$ as independent and identically distributed Gaussian random variables is made for simplicity. More detailed error models can also be used.

*2) Control Design and Parametrization:* We first formulate the constraints and the objective function. Probabilistic versions of the following constraints constitute the normal river operations requirement.

1) $124.65 \le y_{\mathrm{LM},n+1} \le 124.9$, for $i = 1, 2, \ldots, M$.
2) $2500 \le Q_{H,n+i} \le 30\,000$, for $i = 1, 2, \ldots, M$.
3) $-800 \le Q_{H,n+i} - Q_{H,n+i-1} \le 1,200$, for $i = 1, 2, \ldots, M$.

Here $M = 20$ samples (6.67 days) is the prediction horizon. The above constraints are required to be satisfied with probability at least $1 - \epsilon = 0.9$. The upper limit on the change in the flow at Heywoods (1200 ML/day/day) is higher than the lower limit (800 ML/day/day), since the risk of damage to the river banks is higher when the flow decreases. Additionally, the following constraint is required to be satisfied with probability at least $1 - \epsilon_V = 0.99$.

1) $y_{\mathrm{LM},n+i} \le 125$, for $i = 1, 2, \ldots, M$.

The above constraint constitutes the requirement of the flood risk mitigation operation. This constraint is less restrictive than the one above, where the water level should stay below 124.9 mAHD. However, this constraint must be satisfied with a higher probability.

The matrices $\mathbf{Q}$ and $\mathbf{R}$ in the objective function $J(\Gamma_n, \Theta_n)$ [see (12)] were selected such that they prevented water level deviations from the set-point (124.775 mAHD) in Lake Mulwala and minimized the flow release from Hume Reservoir. $\mathbf{Q}$ and $\mathbf{R}$ were tuned based on the experiments on historical data. We used $\mathbf{S} = \mathbf{0}$ [in (12)], since the change in flow at Heywoods is already subjected to constraints.

We used the parametrization of $\Gamma_n$ and $\Theta_n$ matrices as in (8). However, we let the parameters on the subdiagonals (i.e., $\theta_{i+k,j+k}$, where $i > j, k = 1, \ldots, M-1$) of the $\Theta_n$ matrix be the same to reduce the number of optimization variables $d$. Additionally, the parametrization was modified as follows. Due to the system structure, only three elements of the $w_n$ matrix are nonzero. The reason is that most of the states are representing delayed flows leading to update equations of the type $x_{l,n+1} = x_{l+1,n}$ without any uncertainty. Here, $x_l$ and $x_{l+1}$ are the scalar elements of the state vector. Therefore, there are

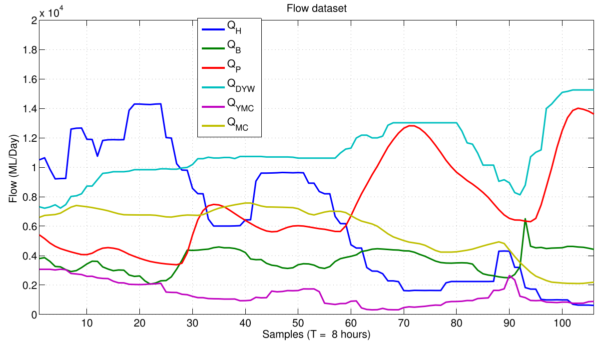


Fig. 7. Inflows and outflows in the data set.

TABLE I
PARAMETERS AND SAMPLE SIZES USED IN THE SIMULATIONS

| $\delta$ | $\beta$ | $\epsilon$ | $\varrho$ | $\epsilon_V$ | $\varrho_V$ | $N_r$ | $N_T$ |
|---|---|---|---|---|---|---|---|
| $1.12 \times 10^{-6}$ | $1.4 \times 10^{-7}$ | 0.1 | 0.05 | 0.01 | 0.005 | 1300 | 8798 |

only three nonzero elements in the $w_n$ vector (corresponding to the water level in Lake Mulwala and the flows at Bandiana and Peechelba). As a consequence, we only need to optimize over the three columns in $\theta_{i,j}$ corresponding to the nonzero elements of the $w_n$ vector. As there is only one control input, the number of parameters is 20 for $\Gamma_n$ and $19 \times 3$ for $\Theta_n$. $d$ was further increased with 1, making $d = 78$. This is because we used a feasibility assurance optimization scheme [8], [28] to solve the scenario problem that requires an additional decision variable. The idea of the scheme is to optimally relax the constraints (by solving an extra optimization problem) before the scenario problem (with the relaxed constraints) is solved (for details see [8] and [28]).

The parameters and sample sizes of the OTI algorithm are given in Table I. For the improving phase, we selected the default solution (see Section II-C) as $Q_{H,d} = 1000$ ML/day, which is nonzero, because some release is necessary for riparian and in-stream environmental needs. The default solution is useful for preventing flooding; however, it violates the minimum flow requirement (2500 ML/day) between Heywoods and Doctors Point (see Fig. 4) and the lower limit on the change of flow at Heywoods ($-800$ ML/day/day) may also be violated. $N_\alpha$ was calculated at each time step of the simulations based on the probability of violations of the solution of the first scenario problem and the default solution against the CC related to flood avoidance, using (30). For computational reasons, we restricted $N_\alpha$ to maximum $N_{\max} = 2500$ in the following simulations.

*3) Simulation Results:* For the simulations, we used the data set from September 19, 2001 to November 16, 2001, which has high inflows from the unregulated rivers ($Q_B$ and $Q_P$). The data set was sampled at $T_s = 8$ h (shown in Fig. 7). All optimization problems were solved by running YALMIP [27] over SDPT3 [53].

Fig. 8 shows the water level in Lake Mulwala. The blue curve shows the controlled water level obtained from the proposed algorithm, the light-blue curve shows the controlled water level obtained by using Step A of the algorithm only (the

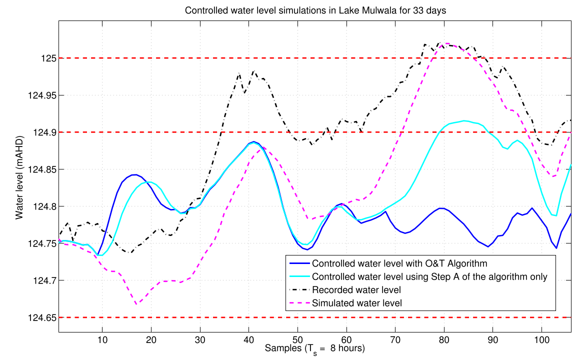


Fig. 8. Controlled water level in Lake Mulwala.

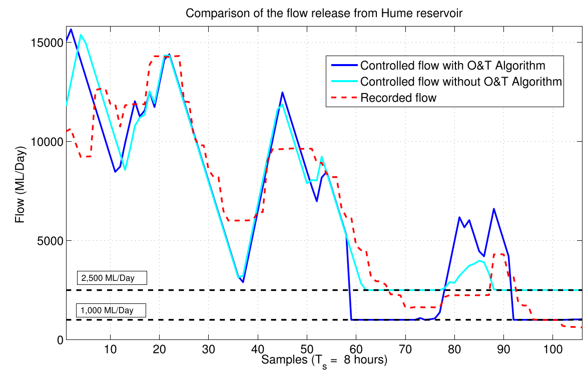


Fig. 9. Regulated flows at Heywoods.

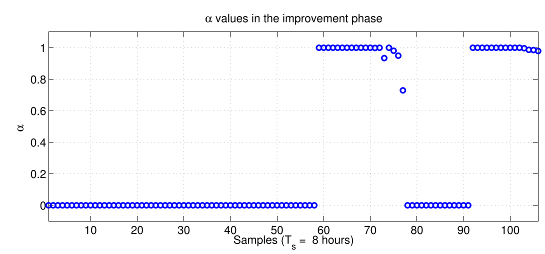


Fig. 10. $\alpha$ values obtained in the improving phase [see Problem (20)].

optimization with the first CC), without considering the risk mitigation CC, the black curve shows the actual recorded water level, and the magenta curve shows the simulation of the model in (35), using the measured input data (in Fig. 7). The model performed reasonably well and picked up the main trends in the data set.

Using the proposed algorithm, the water level was maintained within 124.65 and 124.9 mAHD throughout the simulation (blue curve in Fig. 8). However, it was about to hit the boundary (124.9 mAHD) at the 42nd time instant. It can be explained from the regulated flows at Heywoods. In Fig. 9, the red curve shows the actual recorded flows at Heywoods and the blue curve shows the flow release obtained from the proposed algorithm. From sampling instant 22 to 37, the flows were decreased at the maximum allowed rate, i.e., $-800$ ML/day/day. The restriction on the rate of flow decrease caused the water level to rise, as shown in Fig. 8.

In Fig. 9 (blue curve), the flows at Heywoods went below 2500 ML/day twice. These were the events when the test

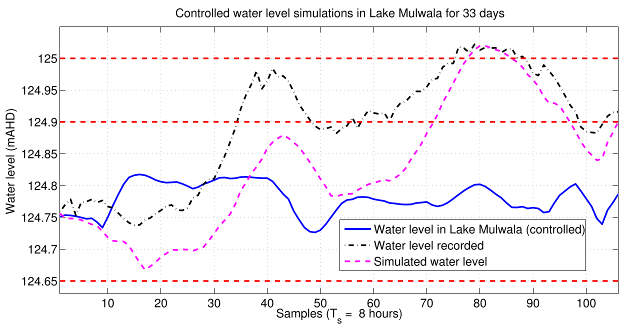


Fig. 11. Controlled water level in Lake Mulwala with nonzero **S** in the objective function.

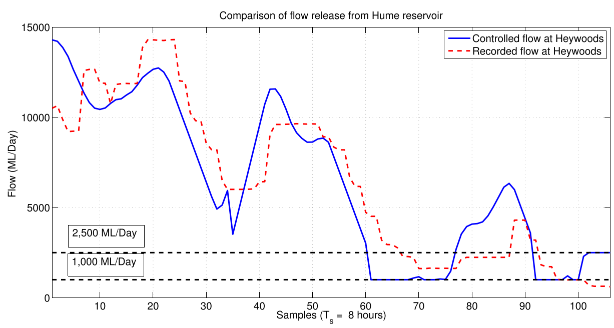


Fig. 12. Regulated flows at Heywoods with nonzero **S** in the objective function.

in Step B-3 of the algorithm (Section IV-A) failed with the obtained solution. In the test, the water level crossed 125 mAHD more than $N_T(\epsilon_V - \varrho_V) = 44$ times out of $N_T = 8798$ different noise scenarios, and the improving procedure was called.

Fig. 10 shows the $\alpha$ values obtained from Problem (20). A nonzero $\alpha$ value indicates an event where an improved solution was needed, and $\alpha = 1$ indicates the instants when the default solution was used, i.e., the flows were reduced to 1000 ML/day. Fig. 10 shows that 65% of the time there was no need to solve Problem (20), and a solution was available at the end of the optimization phase of the algorithm. In the improving phase, $N_\alpha$ exceeded the upper limit ($N_{\max} = 2500$) 42 times in the simulation. The algorithm, however, restricted $N_\alpha$ to be $N_{\max}$ at those steps and the obtained solution was tested against the risk mitigation constraint after the improving was done, and the test passed every time.

In Figs. 8 and 9, the light-blue curves show the controlled water level in Lake Mulwala and the controlled release from Hume Reservoir, respectively, for the case when we only performed Step A of the algorithm. The blue and light-blue curves show a similar behavior in Figs. 8 and 9 until the sampling instants 63 and 58, respectively, which indicates that as long as the testing phase of the algorithm did not report any problems, the response is similar (it is not exactly the same response because the two simulations were run separately and the scenarios were drawn independently in the scenario programs of the stochastic MPC problem). Otherwise, when the algorithm used the default solution, the blue curve shows a stable response around the desired water level (124.775 mAHD), but

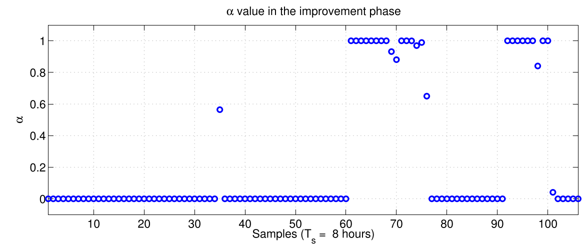


Fig. 13. $\alpha$ values obtained in the improving phase [see Problem (20)] with nonzero **S** in the objective function.

the light-blue curve exceeded the upper limit (124.9 mAHD) from sampling instant 79 to 89.

Fig. 9 shows a number of small frequent changes in the controlled flow release from Hume Reservoir. This can be avoided by considering a nonzero **S** matrix in the objective criterion $J(\Gamma_n, \Theta_n)$ [in (12)], as shown in Figs. 11 and 12. Fig. 12 shows that with this change the control action varied smoothly, as compared with Fig. 9. In this case, we did not experience any rise in the water level close to the 42nd time instant. This can be explained from Fig. 12, where the flows did not get very high around the 22nd sampling instant, as compared with Fig. 9, where it nearly reached 14 500 ML/day. The performance of the algorithm for the rest of the time was similar, and the water level was maintained between its upper and lower limits, as shown in Figs. 8 and 11. Fig. 13 shows the corresponding $\alpha$ values obtained in Problem (20), which indicates that the number of times an improved solution was sought was similar to the previous case (see Fig. 10) where $\mathbf{S} = \mathbf{0}$.

In this section, we have seen that with the application of the OTI algorithm, we managed to keep the water level within safe limits, even when there were large unregulated inflows. The comparison with the recorded data is not completely fair, since we had access to the exact future water demands and we adjusted the flow release every 8 h while the operators only adjusted the flow every 24 h. However, it is still a reasonable comparison because the unregulated in-flows carry the most uncertainty and is the critical factor when it comes to flooding [28]. Furthermore, in the river model, the time delay from Peechelba to Lake Mulwala is 16 h, which is less than 24 h, and therefore, it makes more sense to have the sampling time ($T_s$) of the stochastic MPC problem to be at most 16 h or less. We selected $T_s = 8$ h, because both 16 and 24 are the multiples of 8.

## VI. Conclusion

A control problem for systems that are affected by uncertain inputs and are vulnerable to risks is formulated as an MCCP. The optimization problem includes two CCs: one for normal operations and one for risk avoidance operations, where the latter constraint has a much smaller allowed violation probability. An optimization, testing, and improving-based algorithm to solve the MCCPs has been proposed and applied to the upper part of Murray River in Australia. The simulation results confirmed that the proposed control strategy achieved the control objectives of the river during normal conditions,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



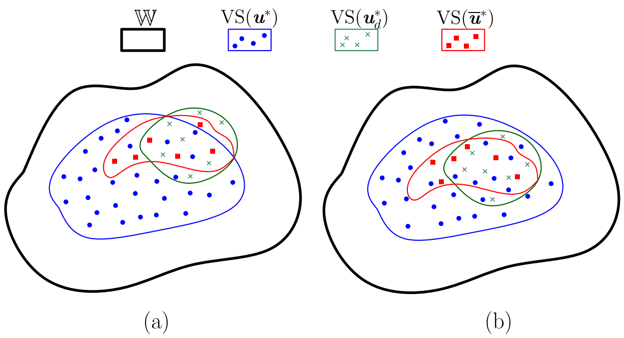


Fig. 14. Pictorial description of violation sets.

and avoided flooding whenever flood risks appeared in a computationally affordable way.

## Appendix
## Proof of Theorem 2

Fig. 14(a) shows the space $\mathbb{W}$ from which the $\mathbf{w}$ realisations are drawn. Let "VS($\mathbf{u}$)" be the set of all realizations $\mathbf{w} \in \mathbb{W}$ that violate the constraint "$g(\mathbf{u}, \mathbf{w}) \leq 0$" evaluated with $\mathbf{u}$. Fig. 14 shows three sets: VS($\boldsymbol{u}^*$), VS($\boldsymbol{u}_d^*$), and VS($\bar{\boldsymbol{u}}^*$) marked with blue circles, green crosses, and red squares, respectively, where $\boldsymbol{u}^*$ is the solution of Problem (15), $\boldsymbol{u}_d^*$ is the default solution, $\bar{\boldsymbol{u}}^* = (1-\alpha^*)\boldsymbol{u}^* + \alpha^*\boldsymbol{u}_d^*$, and $\alpha^*$ is the solution of Problem (20). Note that, VS($\boldsymbol{u}_d^*$) and VS($\bar{\boldsymbol{u}}^*$) are not necessarily contained in VS($\boldsymbol{u}^*$), i.e., the two situations shown in Fig. 14(a) and (b) are both possible.

$\mathcal{T} = \text{VS}(\boldsymbol{u}^*) \setminus \text{VS}(\boldsymbol{u}_d^*)$ [see (29)] is the set that contains all realizations of $\mathbf{w}$ that cause $\boldsymbol{u}^*$ to violate the constraint "$g(\boldsymbol{u}^*, \mathbf{w}) \leq 0$" while $\boldsymbol{u}_d^*$ satisfies the corresponding constraint. In Problem (20), we specifically sample $\mathbf{w}$ from the set $\mathcal{T}$. By convexity, there are no realizations of $\mathbf{w}$ for which the constraint "$g(\mathbf{u}, \mathbf{w}) \leq 0$" is satisfied by both $\boldsymbol{u}^*$ and $\boldsymbol{u}_d^*$ but is violated by $\bar{\boldsymbol{u}}^*$. However, there can be realizations of $\mathbf{w}$ that violate the constraint with $\boldsymbol{u}^*$ and/or $\boldsymbol{u}_d^*$, but satisfy it with $\bar{\boldsymbol{u}}^*$. We can represent VS($\bar{\boldsymbol{u}}^*$) as

$$\text{VS}(\bar{\boldsymbol{u}}^*) = \{\text{VS}(\bar{\boldsymbol{u}}^*) \setminus \mathcal{T}\} \cup \{\text{VS}(\bar{\boldsymbol{u}}^*) \cap \mathcal{T}\}. \tag{38}$$

Again, by the convexity argument above, $\text{VS}(\bar{\boldsymbol{u}}^*) \setminus \mathcal{T} \subseteq \text{VS}(\boldsymbol{u}_d^*)$, and hence

$$\text{VS}(\bar{\boldsymbol{u}}^*) \subseteq \text{VS}(\boldsymbol{u}_d^*) \cup \{\text{VS}(\bar{\boldsymbol{u}}^*) \cap \mathcal{T}\}. \tag{39}$$

Thus,

$$\begin{aligned}\mathbb{P}\big(\text{VS}(\bar{\boldsymbol{u}}^*)\big) &\leq \mathbb{P}\big(\text{VS}(\boldsymbol{u}_d^*)\big) + \mathbb{P}(\text{VS}(\bar{\boldsymbol{u}}^*) \cap \mathcal{T})\\ &= \mathbb{P}\big(\text{VS}(\boldsymbol{u}_d^*)\big) + \mathbb{P}(\text{VS}(\bar{\boldsymbol{u}}^*)|\mathcal{T})\mathbb{P}(\mathcal{T})\\ &\leq \hat{\hat{\epsilon}}_d + \mathbb{P}(\text{VS}(\bar{\boldsymbol{u}}^*)|\mathcal{T})\hat{\hat{\epsilon}}_T \end{aligned} \tag{40}$$

where we have used the upper bounds $\mathbb{P}(\text{VS}(\boldsymbol{u}_d^*)) \leq \hat{\hat{\epsilon}}_d$ and $\mathbb{P}(\mathcal{T}) \leq \hat{\hat{\epsilon}}_T$. As $N_\alpha$ satisfies (16) with $\epsilon = \epsilon_\alpha$ and $d = 1$, the solution to Problem (20) satisfies $\mathbb{P}(\text{VS}(\bar{\boldsymbol{u}}^*)|\mathcal{T}) \leq \epsilon_\alpha$ with a confidence $1-\beta$. Using $\epsilon_\alpha \leq (\epsilon_V - \hat{\hat{\epsilon}}_d)/\hat{\hat{\epsilon}}_T$, the theorem follows from (40).

## Acknowledgment

The authors would like to thank Murray Darling Basin Authority, Australia, for providing the operational Murray River data and for discussions on the river operations.

## References


[1] J. M. Maciejowski, *Predictive Control: With Constraints*. London, U.K.: Prentice-Hall, 2002.

[2] I. Batina, "Model predictive control for stochastic systems by randomized algorithms," Ph.D. dissertation, Techn. Univ. Eindhoven, Eindhoven, The Netherlands, 2004.

[3] M. Cannon, B. Kouvaritakis, and X. Wu, "Probabilistic constrained MPC for multiplicative and additive stochastic uncertainty," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1626–1632, Jul. 2009.

[4] E. Cinquemani, M. Agarwal, D. Chatterjee, and J. Lygeros, "Convexity and convex approximations of discrete-time stochastic control problems with constraints," *Automatica*, vol. 47, no. 9, pp. 2082–2087, 2011.

[5] J. A. Primbs and C. H. Sung, "Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 221–230, Feb. 2009.

[6] G. Schildbach, L. Fagiano, and M. Morari, "Randomized solutions to convex programs with multiple chance constraints," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2479–2501, 2013.

[7] H. A. Nasir and E. Weyer, "Control of a river stretch with uncertain inflows," in *Proc. 4th Austral. Control Conf. (AUCC)*, Nov. 2014, pp. 37–42.

[8] H. A. Nasir, S. Garatti, and E. Weyer, "Scenario based stochastic MPC schemes for rivers with feasibility assurance," in *Proc. Eur. Control Conf. (ECC)*, 2016, pp. 1928–1933.

[9] M. Prandini, S. Garatti, and J. Lygeros, "A randomized approach to stochastic model predictive control," in *Proc. 51st Conf. Decision Control (CDC)*, Dec. 2012, pp. 7315–7320.

[10] G. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[11] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 257–280, 2011.

[12] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annu. Rev. Control*, vol. 33, no. 2, pp. 149–157, 2009.

[13] P. Patrinos, S. Trimboli, and A. Bemporad, "Stochastic MPC for real-time market-based optimal power dispatch," in *Proc. Decision Control Eur. Control Conf. (CDC-ECC)*, Dec. 2011, pp. 7111–7116.

[14] D. P. De Farias and B. Van Roy, "On constraint sampling in the linear programming approach to approximate dynamic programming," *Math. Oper. Res.*, vol. 29, no. 3, pp. 462–478, 2004.

[15] E. Erdoğan and G. Lyengar, "Ambiguous chance constrained problems and robust optimization," *Math. Programm.*, vol. 107, nos. 1–2, pp. 37–61, 2006.

[16] A. Nemirovski and A. Shapiro, "Scenario approximations of chance constraints," in *Probabilistic and Randomized Methods for Design under Uncertainty*. London, U.K.: Springer, 2006, pp. 3–47.

[17] G. C. Calafiore, "Random convex programs: Dealing with the non-existing solution nuisance," in *Proc. 49th Conf. Decision Control (CDC)*, Dec. 2010, pp. 1939–1944.

[18] A. Carè, S. Garatti, and M. C. Campi, "Scenario min-max optimization and the risk of empirical costs," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2061–2080, 2015.

[19] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[20] Y. Oishi, "Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities," *Automatica*, vol. 43, no. 3, p. 538–545, 2007.

[21] T. Alamo, R. Tempo, and E. F. Camacho, "Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2545–2559, Nov. 2009.

[22] T. Alamo, R. Tempo, A. Luque, and D. R. Ramirez, "Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms," *Automatica*, vol. 52, pp. 160–172, Feb. 2015.

[23] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q. G. Wang, "Sequential randomized algorithms for convex optimization in the presence of uncertainty," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2565–2571, Sep. 2016.

[24] A. Carè, S. Garatti, and M. C. Campi, "FAST—Fast algorithm for the scenario technique," *Oper. Res.*, vol. 62, no. 3, pp. 662–671, 2014.

[25] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[26] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

[27] J. Lofberg, "Approximations of closed-loop minimax MPC," in *Proc. 42nd Conf. Decision Control (CDC)*, vol. 2. Dec. 2003, pp. 1438–1442.

[28] H. A. Nasir, "System identification and control of rivers," Ph.D. dissertation, Dept. Elect. Electron. Eng., Univ. Melbourne, Parkville, VIC, Australia, 2016.

[29] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: SIAM, 1994.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[31] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, nos. 3–4, pp. 231–357, 2015.

[32] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1211–1230, 2008.

[33] G. C. Calafiore, "Random convex programs," *SIAM J. Optim.*, vol. 20, no. 6, pp. 3427–3464, 2010.

[34] X. Zhang, S. Grammatico, G. Schildbach, P. Goulart, and J. Lygeros, "On the sample size of random convex programs with structured dependence on the uncertainty," *Automatica*, vol. 60, pp. 182–188, Oct. 2015.

[35] X. Zhang, K. Margellos, P. Goulart, and J. Lygeros, "Stochastic model predictive control using a combination of randomized and robust optimization," in *Proc. 52nd Annu. Conf. Decision Control (CDC)*, Dec. 2013, pp. 7740–7745.

[36] X. Zhang, A. Georghiou, and J. Lygeros, "Convex approximation of chance-constrained MPC through piecewise affine policies using randomized and robust optimization," in *Proc. 54th Annu. Conf. Decision Control (CDC)*, Dec. 2015, pp. 3038–3043.

[37] N. Kariotoglou, K. Margellos, and J. Lygeros, "On the computational complexity and generalization properties of multi-stage and stage-wise coupled scenario programs," *Syst. Control Lett.*, vol. 94, pp. 63–69, Aug. 2016.

[38] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. Int. Symp. Comput. Aided Control Syst. Design*, Sep. 2004, pp. 284–289.

[39] M. Grant and S. Boyd. (Sep. 2013). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.0 Beta*. [Online]. Available: http://cvxr.com/cvx

[40] A. Prékopa, *Stochastic Programming*, vol. 324. Amsterdam, The Netherlands: Springer, 2013.

[41] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM J. Optim.*, vol. 17, no. 4, pp. 969–996, 2006.

[42] H. A. Nasir, A. Carè, and E. Weyer, "A randomised approach to flood control using Value-at-Risk," in *Proc. 54th Conf. Decision Control (CDC)*, Dec. 2015, pp. 3939–3944.

[43] S. Garatti and M. C. Campi, "Modulating robustness in control design: Principles and algorithms," *IEEE Control Syst. Mag.*, vol. 33, no. 2, pp. 36–51, Apr. 2013.

[44] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Statist. Sci.*, vol. 16, no. 2, pp. 101–117, 2001.

[45] M. Foo, S. K. Ooi, and E. Weyer, "Modelling of rivers for control design," in *System Identification, Environmental Modelling, and Control System Design*. London, U.K.: Springer-Verlag, 2012, pp. 423–447.

[46] M. Foo, S. K. Ooi, and E. Weyer, "System identification and control of the Broken River," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 618–634, Mar. 2014.

[47] X. Litrico, "Robust IMC flow control of SIMO dam-river open-channel systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 3, pp. 432–437, May 2002.

[48] H. A. Nasir and E. Weyer, "System identification of the upper part of Murray River," *Control Eng. Pract.*, vol. 52, pp. 70–92, Jul. 2016.

[49] P. C. Young, A. Castelletti, and F. Pianosi, "The data-based mechanistic approach in hydrological modelling," in *Topics on System Analysis and Integrated Water Resources Management*. Amsterdam, The Netherlands: Elsevier, 2007, pp. 27–48.

[50] M. Papageorgiou and A. Messmer, "Flow control of a long river stretch," *Automatica*, vol. 25, no. 2, pp. 177–183, 1989.

[51] C. Setz, A. Heinrich, P. Rostalski, G. Papafotiou, and M. Morari, "Application of model predictive control to a cascade of river power plants," in *Proc. IFAC World Congr.*, 2008, pp. 11978–11983.

[52] M. F. L. Foo, "Modelling and control design of river systems," Ph.D. dissertation, Dept. Elect. Electron. Eng., Univ. Melbourne, Parkville, VIC, Australia, 2011.

[53] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "DPT3—A MATLAB software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 545–581, 1999.

**Hasan Arshad Nasir** received the bachelor's degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2009, the master's degree in computer engineering from the Lahore University of Management Sciences, Lahore, Pakistan, in 2011, and the Ph.D. degree from The University of Melbourne, Melbourne, VIC, Australia, in 2016.

He is currently a Post-Doctoral Research Fellow with the Department of Electrical and Electronic Engineering, The University of Melbourne. During his Ph.D., he also visited the research groups at Politecnico di Milano, Milan, Italy, and the Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary. His current research interests include system identification and optimization-based control.

**Algo Carè** received the Ph.D. degree in informatics and automation engineering from the University of Brescia, Brescia, Italy.

He is currently a Research Fellow with the Department of Information Engineering, University of Brescia. He spent two years at The University of Melbourne, Melbourne, VIC, Australia, as a Research Fellow in system identification with the Department of Electrical and Electronic Engineering.

Dr. Carè was a recipient of a two-year ERCIM Fellowship in 2016 that he spent at the Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Budapest, Hungary, and at the Multiscale Dynamics Group, National Research Institute for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. He received the triennial Stochastic Programming Student Paper Prize by the Stochastic Programming Society for the period 2013–2016. His current research interests include data-driven decision methods, system identification, and learning theory.

**Erik Weyer** received the Siv. Ing. and Ph.D. degrees from the Norwegian Institute of Technology, Trondheim, Norway.

From 1994 to 1996, he was a Research Fellow with The University of Queensland, Brisbane, QLD, Australia, before he joined The University of Melbourne in 1997. He has held visiting positions at the University of Brescia, Brescia, Italy, the Technical University of Vienna, Vienna, Austria, and Politecnico di Milano, Milan, Italy. He is currently a Professor with the Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia. His current research interests include system identification and control.

Dr. Weyer received the IEEE Control Systems Technology Award in 2014 for the development and implementation of controls for irrigation channels and water management. He was an Associate Editor of the IEEE TRANSACTIONS OF AUTOMATIC CONTROL from 2010 to 2012. He is currently an Associate Editor of AUTOMATICA.