Compositional Control Synthesis for Water Management System*

Esther H. Kim[1] and Kim G. Larsen[1] and Martijn A. Goorden[1] and Thomas D. Nielsen[1]

***Abstract*— The increased frequency and severity of extreme weather events are challenging traditional static control strategies for stormwater detention ponds, which are critical components in urban water management infrastructures. This paper introduces a compositional control methodology, rooted in formal verification and reinforcement learning, and tailored for the synthesis of a joint optimal control strategy for the management of distributed but interconnected ponds. Combining hybrid Markov Decision Processes (HMDPs) and reinforcement learning via Uppaal Stratego, the compositional control strategy provides a balance between fully centralized and decentralized control strategies, both in terms of quality and computational complexity. Based on a real-world case study we analyze and compare the proposed methodology and show how the synthesized strategies can control the timing and volume of water discharge, reducing the risk of overflow caused by the simultaneous discharge of rainwater collected in multiple ponds.**

I. Introduction

Complex Cyber-Physical Systems (CPSs) integrate computational and physical processes, making their management and control intricate [1]. For hybrid systems, controller synthesis is further complicated by the coexistence of continuous and discrete dynamics [2]. Traditional centralized controllers are becoming increasingly complex and inefficient for such advanced CPS architectures. This underscores the pressing need for innovative approaches that address the nuanced challenges posed by CPSs and hybrid system's controller design while bypassing the inefficiencies inherent in centralized control methods.

In recent times, compositional control has emerged as an effective solution for complex system management. Central to this approach are concepts like assume-guarantee reasoning and contract-based design [3]. By defining interactions between subsystems through formal assumptions and guarantees, these methodologies offer a modular perspective to control synthesis [4]. The exact nature of these formalisms varies, depending on the overarching system specifications [5]. Such interactions can often be represented using automata or temporal logic formulas. An example of this method is in an online compositional synthesis for a multi-room floor-heating system, significantly reducing control complexity [6]. Advancements in this domain include the development of safety controllers for systems with linear dynamics and interdependent parameters, enhancing robustness through the formulation of controlled invariant sets, with demonstrated effectiveness in vehicular safety applications [7].

Applying the principles of compositional control within the realm of environmental engineering reveals a promising avenue for improving the management of stormwater detention ponds [8]. These stormwater detention ponds are pivotal in reducing flood risks and enhancing water quality. Yet, the prevailing control mechanisms for managing the discharge of water, largely dependent on static hydraulic structures, are proving inadequate in the face of shifting weather patterns and varying flow conditions brought about by climate change. The imperative for dynamic and flexible control systems is thus magnified, particularly in the case of interlinked detention pond networks that converge into a single watercourse. In these systems, the control strategy must be sufficiently advanced to reduce the ecological impact on the stream and prevent overflow from stormwater detention ponds. Given their integral role in urban infrastructure, these systems can significantly benefit from the modularity and adaptive capabilities of such control strategies, positioning them as exemplary applications of responsive control mechanisms.

Real-time control (RTC) enhances the efficiency and performance of urban water management systems by incorporating current sensor readings into both traditional rule-based frameworks and more dynamic control strategies. By enabling rapid and adaptive control decisions, RTC ensures the optimal use of storage and transport capacities within urban water infrastructures, adjusting to varying conditions in real time [9]. Furthermore, RTC combined with machine learning has surfaced as a pivotal asset, especially in stormwater control. It provides an efficient computational alternative to the popular model predictive control in real-time settings [10].

Recent advancements in stormwater control strategies have brought attention to the application of reinforcement learning (RL) for flood mitigation and sediment treatment for reducing water pollution caused by surface run-off [11]. When contrasted with traditional local rule-based controls, RL aims to find a balance between flood mitigation and sediment treatment. The local controls may unintentionally elevate the risk of flooding. RL holds promise for providing a solution that prioritizes equilibrium and can address multiple objectives simultaneously [12].

Multi-Agent Reinforcement Learning (MARL) provides another solution for decentralized control. MARL studies, such as those applied in wastewater treatment [13] and robotic controls [14], adhere to a centralized training yet de-

*This research is funded by the Digital Research Centre Denmark (DIREC), Innovation Fund Denmark and the Villum Investigator Grant S4OS.

[1]Esther H. Kim, Kim G. Larsen, Martijn A. Goorden, and Thomas D. Nielsen are with the Department of Computer Science, Aalborg University, 9220 Aalborg East, Denmark {hki, kgl, mgoorden, tdn}@cs.aau.dk

centralized execution framework. While centralized setups have demonstrated efficacy in reducing urban flooding [15], [16], their reliance on expansive real-time data transfers might expose them to communication setbacks [17]. On the other hand, decentralized systems avoid a central command structure, enabling agents to operate local actuators independently. While this reduces disruptions related to communication, it may introduce coordination challenges among subsystems in areas such as catchments [18].

In this paper, we present a novel framework for the management of interconnected stormwater detention ponds, modeled by a hybrid Markov Decision Process (HMDP) [19]. Our focus is on a system that integrates three detention ponds and a connecting stream within a compositional control scheme. Compositional control ensures the individual and collective management of interconnected systems, while at the same time reducing the computational complexity when compared to a fully centralized strategy. The interconnected control strategy not only facilitates independent response actions by each pond to its immediate conditions but also aligns these actions with the collective goal of optimal water system management. Based on a real-world use case we compare centralized, decentralized, and compositional control strategies. The results from the analysis show that the compositional approach represents the best of both worlds, in the sense of being significantly more computationally efficient than a fully centralized approach while at the same time retaining the control qualities of this approach. The principles underlying the compositional learning approach are not specific to the water management domain and would therefore also be applicable in other domains that can be expressed as a compositional system.

This paper is structured as follows. Section II introduces stormwater detention ponds in more detail. Section III provides the preliminaries for this paper. Then, Section IV introduces our new compositional control framework, which is compared to the centralized and decentralized control frameworks. In Section VI experimental results are presented that showcase the performance of the proposed compositional control framework, and in Section V details the implementation specifics of ou experiments. Finally, Section VII concludes the paper.

II. Case study: Managing multiple stormwater detention ponds

In this case study, we elucidate the challenges posed by the current static management of water streams and highlight the need for more dynamic and integrated approaches. Our focus is on the Giber Å stream, a significant 12-kilometer long waterway that gracefully meanders through the suburban regions of Aarhus, Denmark (see Figure 1). Along this stream, several stormwater detention ponds have been placed to capture rain runoff originating from the neighboring urban areas (see Figure 2 for an overview). Our focus is on three such ponds located in the Vilhelmsborg vicinity of Giber Å. Especially during periods of heavy precipitation, these stormwater detention ponds play a crucial role in delaying the direct discharge of rain runoff that could potentially cause flooding in the stream, thereby preventing such overflow.

Fig. 1: Satellite representation of the Giber Å stream and adjacent stormwater detention ponds in Aarhus, Denmark. The blue line delineates the trajectory of the stream, and the yellow markers indicate the positions of the ponds. Image sourced from Google Earth.

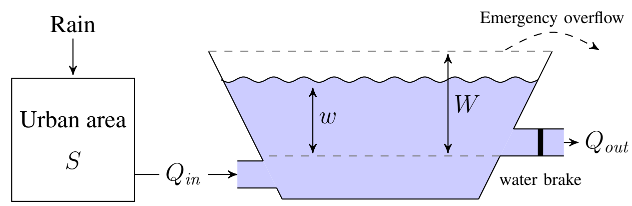

Fig. 2: Simplified overview of a stormwater detention pond.

A. Static Management of Giber Å Stream

Currently, these stormwater detention ponds operate under a fixed control framework, where the discharge rates remain constant [20]. Furthermore, the ponds operate independently, not taking into account the discharge rates of the other ponds nor the condition of the stream into which the water from the ponds is being discharged. Such segregated and static modes of operation cannot dynamically manage outflows in response to actual and predicted rainfall.

B. Control Objectives and Modeling Background

Excessive discharge from these ponds not only causes overflow in the stream but also results in rapid fluctuations in biological indicators such as temperature, acidity, and oxygen levels, thereby disrupting the ecological balance.

The primary goal of this research is to prevent overflows from the stream and the ponds and minimize the discharge of pollutants to the stream. This entails controlling the water released from the detention ponds, with the objective of maintaining a high water level (to ensure sedimentation of pollutants) while at the same time avoiding overflow in the ponds and the stream.

In the use case, each pond has an average pond area of A_p = 706.9 m^2 and is designed to withstand a maximum water level of W = 2 m. A circular orifice, situated at the base of the ponds with a diameter of 30 cm, is employed with three controller settings — fully open, half open, and fully closed — chosen on an hourly basis, resulting in outflow Q^i_{out}, where i denotes the pond. The inflow to each stormwater detention pond (Q^i_{in}) consists of rain runoff from its connected urban catchment (UC^i). Each of these catchments consists of a 40-hectare area with an impervious rate of 25%, indicating that a quarter of the catchment area is covered with impervious surfaces such as pavement and buildings. Impervious surfaces prevent water from infiltrating into the ground, leading to increased surface runoff and reduced groundwater recharge.

For modeling the use case, we adopt the Hybrid Markov Decision Process (HMDP) framework, based on which control strategies will be synthesized. The following sections will elaborate on the HMDP model and its application in achieving our ecological preservation goals for the Giber Å stream.

III. Preliminaries

A. Hybrid Markov Decision Process

We utilize the mathematical framework of the Hybrid Markov Decision Process (HMDP) to model compositional interactions between models [19].

Definition 1: A *Hybrid Markov Decision Process* $\mathcal{M}$ is a tuple (C, U, X, F, δ), where:

- The controller C is a finite set of (controllable) modes $C = \{c_1, \ldots, c_k\}$.
- The uncontrollable environment U is a finite set of (uncontrollable) modes $U = \{u_1, \ldots, u_l\}$.
- $X = \{x_1, \ldots, x_n\}$ is a finite set of continuous (real-valued) variables with $x : X \to \mathbb{R}$ being a valuation.
- For each $c \in C$ and $u \in U$, $F_{c,u} : \mathbb{R}_{>0} \times \mathbb{R}^X \to \mathbb{R}^X$ describes the flow-function, which is the evolution of the continuous variables over time in the combined mode (c, u).
- δ is a family of probability functions $\delta_\gamma : \mathbb{R}_{\geq 0} \times U \to [0, 1]$, where $\gamma = (c, u, x)$ is a global configuration. $\delta_\gamma(\tau, u')$ is the probability that in the global configuration $\gamma = (c, u, x)$ uncontrollable mode u will change to u' after a time delay τ. Note that $\sum_{u'} \int_\tau \delta_\gamma(\tau, u')\mathrm{d}\tau = 1$. This ensures that the probabilities across all possible u' sum up to 1 for any given time delay τ.

The set of global configurations $C \times U \times \mathbb{R}^X$ is denoted by $\mathbb{C}$.

Conceptually, the uncontrollable modes, such as the duration and intensity of rain events, are encapsulated within the uncontrollable environment variable U. The flow function F models the dynamics of the water level in the pond, which is represented by the continuous real-valued state variables X. The stochastic behavior of weather patterns is embedded within the probability function δ. The specific orifice setting, which determines how water is released, is represented by the controller C. While this controller can potentially have various modes of operation, in this paper, we focus on a specific scenario. We limit the controller's functionality to switch modes only at a predetermined frequency. This is characterized by a switching period $P \in \mathbb{R}_{\geq 0}$.

Note that Definition 1 above supports several HMDP modeling formalisms, abstracting away any domain structure. For instance, the uncontrollable modes can be seen as an enumeration of the state space defined by a collection of random variables modeling rain events. Domain structure is instead given implicitly in the specification of the flow and probability functions.

The composition of two HMDPs $\mathcal{M}_1$ and $\mathcal{M}_2$, denoted by $\mathcal{M} = \mathcal{M}_1 \parallel \mathcal{M}_2$, is also an HMDP. The structure of the resulting model $\mathcal{M}$, as reflected by the composition, is again captured by the flow and probability functions of $\mathcal{M}$, which should be consistent with the corresponding functions in $\mathcal{M}_1$ and $\mathcal{M}_2$ (e.g., the models should define the same probability distribution over any shared modes).

B. HMDP strategies

A *strategy* for an HMDP $\mathcal{M}$ is a function $\sigma : \mathbb{C} \to 2^C$ that, given a configuration $\gamma = (c, u, x)$, returns a set of controllable modes.

For a given strategy σ and HMDP $\mathcal{M}$, the model is again an HMDP, denoted by $\mathcal{M} \upharpoonright \sigma$; if σ defines a singleton set for any configuration γ of $\mathcal{M}$, then $\mathcal{M} \upharpoonright \sigma$ reduces to a hybrid Markov process. For a compositional model $\mathcal{M} = \mathcal{M}_i \parallel \mathcal{M}_j$, a strategy σ_i for $\mathcal{M}_i$ can vacuously be extended to a strategy for $\mathcal{M}$. In this case, σ_i is said to be a *partial strategy* for $\mathcal{M}$ and defines an HMDP $\mathcal{M} \upharpoonright \sigma_{_i}$, where the controllable modes of $\mathcal{M}_1$ are constrained according to σ_i.

A run of an HMDP $\mathcal{M}$ under strategy σ is a sequence of transitions between configurations γ_i, where the control modes change according to σ.

We use an objective function f to assess the effectiveness of this strategy σ. The function f takes a global configuration from $\mathbb{C}$ and returns a real number. To evaluate σ, we calculate its expected value over all possible runs of $\mathcal{M} \upharpoonright \sigma$. We start from an initial configuration γ and consider a time horizon H, which is any non-negative real number. Mathematically, this is captured by $\mathbb{E}^{\mathcal{M},\gamma}_{\sigma,H}(f)$.

C. UPPAAL STRATEGO

We employ the modeling tool UPPAAL STRATEGO [21] for synthesizing optimal control strategies within a specified environment. Notably, learning algorithms, such as Q-learning, have been integrated into UPPAAL STRATEGO, as previously described [22].

For Q-learning, UPPAAL STRATEGO performs sample runs from the HMDP model $\mathcal{M}$ to compute Q-values. These calculated values are subsequently employed to refine the strategy. This iterative process is executed with the updated strategy until there is a convergence in performance.

For continuous state spaces UPPAAL STRATEGO resort to an online partition refinement techniques for discretizing the state space [22]. This technique enables the tool to dynamically refine its representation of the state space during exploration. Consequently, it becomes feasible to derive effective strategies without the need for exhaustive state analysis.

In the model $\mathcal{M}$ each mode in the controller corresponds to a particular strategy or set of operations applied across the ponds. The uncontrollable environment (U) encapsulate factors like weather forecasts, which are inherently unpredictable and uncontrollable. The continuous (real-valued) variables (X) represent sensor-based obervations, like the observed water level in the ponds and stream. The flow-function (F) models the dynamics of the water level in the ponds and stream over time, given a specific mode of operation for the controller and a specific uncontrollable environmental condition. The probability function (δ) captures the inherent uncertainty in predicting the change in weather conditions after a specific duration. A full specification of the model can be found in [15].

IV. Compositional Control Synthesis

A. Control problem definition

The primary objective of the controller is to ensure the safety of both the stream and the ponds by effectively regulating the discharge from the pond. In this context, the safety of the stream is defined as preventing overflow and minimizing the sudden influx of discharge from the stormwater detention ponds. The junctions, where water discharges from each pond, are located at different positions along the stream. The observation point for the stream's water level w_s is downstream from the ponds. Without overflowing, we aim to follow a desired target level w_s^{target}. The constant variable α has been used to penalize overflow (e.g., 10,000). Formally,

$$\frac{\mathrm{d}cost_s}{\mathrm{d}t} = \begin{cases} \alpha \times (Q^s_{in} - Q^s_{out}) & \text{if } (w_s \geq W_s), \\ \left(1 - \frac{w_s}{w_s^{target}}\right)^2 & \text{otherwise.} \end{cases}$$

For our purposes, the safe operation of stormwater detention ponds is defined as preventing overflow. An uncontrolled overflow can have cascading consequences, affecting the stream and the surrounding environment. Beyond the safety objective, the secondary goal for stormwater detention ponds is to maximize the sedimentation of pollutants originating from urban catchment runoff. The sedimentation rate is directly proportional to the volume of water in the pond: the more water it contains, the higher the sedimentation of particles.

We quantify the overflow amount by assessing the water level w_i of pond i, a continuous variable ranging between 0 and W_i, which denotes the maximum depth of the pond. Without overflowing, we aim to maximize particle sedimentation. Formally,

$$\frac{\mathrm{d}cost_i}{\mathrm{d}t} = \begin{cases} \alpha_i \times (Q^i_{in} - Q^i_{out}) & \text{if } (w_i \geq W_i), \\ \left(1 - \frac{w_i}{W_i}\right) & \text{otherwise.} \end{cases}$$

TABLE I: Notations for Variables Used in Controller Synthesis.

Variable	Description
O_t^{rain}	observation of rain intensity at time step t
O_t^{st}	observed water level of the stream at time step t
O_t^i	observed water level of pond i at time step t
C_t^i	control decision of pond i at time step t
σ_t^i	control strategy of pond i at time step t

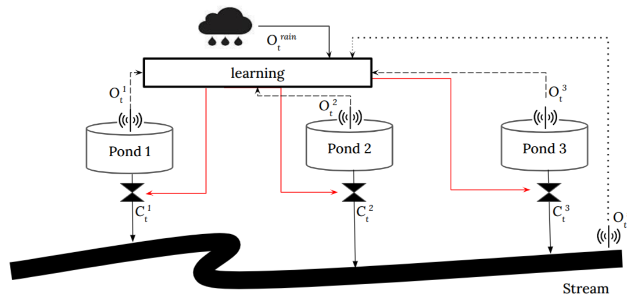

Fig. 3: Overview of centralized control synthesis. Centralized control synthesis observes the water levels of all ponds, the water level of the stream, and the rain forecast.

B. Methodology

In our study, we delve into three primary control architectures: a centralized control system, a decentralized control system, and our proposed compositional control system. Each architecture offers a distinct approach to managing and controlling systems modeled as HMDPs, considering optimization objectives. The notations used in the figures in this section are described in Table I.

In centralized control, a single control unit can control the outflow of the three ponds of our case study, see Figure 3. Therefore, we use one HMDP $\mathcal{M}_{\text{cen}}$ that captures the uncontrollable inputs U as well as the controllable outputs C of all ponds. Furthermore, everything (water levels of all ponds and the stream water level) is observable to the centralized controller. Therefore, a single control strategy is obtained with reinforcement learning. Centralized control corresponds to our previous work [15].

To learn a centralized controller, we formulated the following cost function $Cost_{cen}$ as the control objective to be minimized:

$$Cost_{cen} = cost_s + \Sigma_{i=1}^{n} cost_i$$

In this equation, the cost function of the centralized controller is the combination of the cost of the stream $cost_s$ and the cost of each pond $cost_i$.

In decentralized control, a single controller is synthesized for each pond individually *without* taking the presence of the other ponds into account, see Figure 4. So a controller for pond i is synthesized based on just the observations of pond i and the stream, together with the weather forecast.

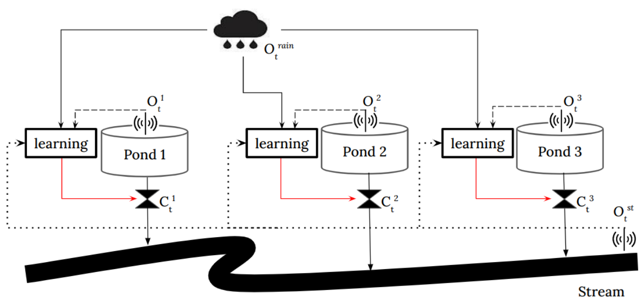

Fig. 4: Overview of decentralized control synthesis. Each pond observes the water level of only that pond, the water level of the stream, and the rain forecast. Utilizing this data, a controller is synthesized for each pond separately.

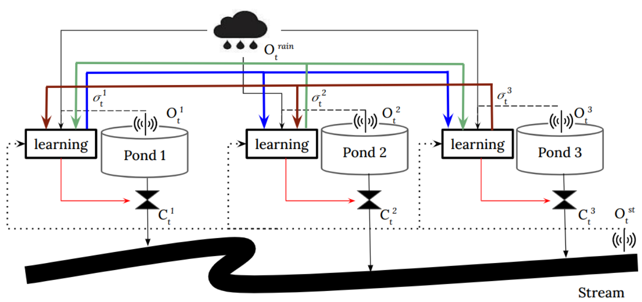

Fig. 5: Overview of compositional control synthesis. Each pond observes the water level of only that pond, the water level of the stream, and the rain forecast. Furthermore, the control strategies of the other ponds are available during learning.

No information from other ponds or their controllers is used. Therefore, we use three HMDPs $\mathcal{M}_{decen,i}$, one for each pond.

To learn a decentralized controller for pond i, we formulated the following cost function $Cost_{decen,i}$ as control objective to be minimized:

$$Cost_{decen,i} = cost_s + cost_i$$

This cost function reflects that each controller only takes its own pond into account as well as the shared stream. Therefore, ponds share water level information indirectly via the shared stream. For example, if a pond discharges a large volume of water, the increasing water level in the stream will eventually be detected by the other controllers.

Operating within this decentralized architecture, agents or subsystems function autonomously, relying solely on localized observations for decision processes. This design inherently fosters scalability, but decentralized choices, which might be locally optimal, might not align with global optimal strategies. Moreover, the intrinsic independence of each agent can occasionally cause conflicts, potentially instigating inefficiencies in the overarching system.

In compositional control, a single controller is synthesized for each pond individually, similar to decentralized control, but now taking the initial water levels and the control strategies of the other ponds into account, see Figure 5. For example, while learning a control strategy for pond 1, the control strategy of pond 2 is considered to be an uncontrollable input such that the future discharge of pond 2 into the shared stream can be taken into account. Therefore, we use three HMDPs $\mathcal{M}_{comp,i}$, one for each pond, that all include all ponds and the stream, but for $\mathcal{M}_{comp,i}$ only the output setting of the orifice of pond i is marked as controllable, while those of the other ponds are marked as uncontrollable.

To learn a compositional controller for pond i, we formulated the following cost function $Cost_{comp,i}$ as control objective to be minimized:

$$Cost_{comp,i} = cost_s + cost_i$$

Note that $Cost_{comp,i} = Cost_{decen,i}$.

Initially, when synthesizing a compositional strategy for pond i, the strategies of the other ponds may not yet be synthesized. Therefore, some temporarily initial strategy needs to be assumed, which could be, e.g., a fixed strategy or a purely random strategy. After synthesizing compositional controllers for each stormwater pond, the process could be repeated using the just synthesized controllers as input. This should provide reinforcement learning a more accurate view of how the other controllers might make decisions in the near future. In general, this process might not terminate, as there is no guarantee that a fixed-point is reached. Therefore, we currently only repeat this process a fixed number of times every controller switching period P.

C. Online control

To address swift environmental fluctuations, we incorporated model-predictive control [23]. With this approach, every hour the controller is re-synthesized using the latest sensor measurements of the true water levels in the system as well as a new 6-hour weather forecast.

For compositional control, we bootstrap the first synthesis iteration by using the controllers obtained from the previous control period. These previously synthesized controllers might provide a better initial control strategy for the uncontrollable ponds than a static or random control strategy.

V. Implementation

In our experiment, the decentralized control system is composed of three individual UPPAAL system models $\mathcal{M}^i_{decen}$, each modeling a single pond and the shared stream. For each system model, we synthesize an optimal control strategy $\sigma^i_{t,H}$ for each pond i and time horizon H, considering only observations of the specific pond and stream as well as the weather forecast.

The compositional control system is composed of three individual UPPAAL system models $\mathcal{M}^i_{compo}$. Compared to the decentralized model, each $\mathcal{M}^i_{compo}$ model represents all three ponds and the shared stream (see Figure 6). For

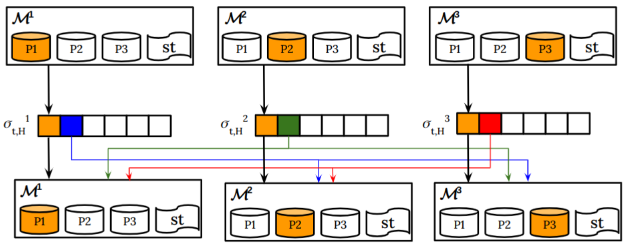

Fig. 6: Illustration depicting the process of compositional control synthesis from a time step perspective.

each system model $\mathcal{M}^i_{compo}$, an optimal control strategy $\sigma^i_{t,H}$ of each pond i is synthesized, considering observations of the specific pond and stream, weather forecast, and the previously synthesized startegies for the other ponds (a fixed control strategy is assumed for the initial time step).

We implemented this as an online control method using the STOMPC framework [24], which integrates synthesis and co-simulation based on the MPC scheme. For our work, UPPAAL STRATEGO is utilized for synthesis, while PySWMM [25] is employed for co-simulation.

VI. Experimental results

The primary objective is to synthesize a control strategy that minimizes the expected cost, as delineated by the cost function detailed in Section IV. Specifically, our analysis focuses on the performance of controller synthesis under varying synthesis conditions. We take into account the effects of

- the overflowing quantity from both ponds and the stream,
- the sedimentation in each pond, and
- deviations in the desired water level of the stream.

For modeling the rainfall in the use case, we utilized historical rain data from September 10, 2019, from 00:00 to 12:00 (for 12 hours). Additionally, we incorporated an uncertainty factor, $\varepsilon = 10\%$, to the observed interval durations and rain intensities to simulate a weather forecast. This data was sourced from the Danish Meteorological Institute (2020). Particularly, this period was marked by intense and concentrated rainfall, making it challenging for both ponds and a stream to avoid overflow.

Figure 7 shows qualitative experimental results of compositional control synthesis. The rain and urban catchment both appear at the top of the plot because the rain that enters the urban catchments eventually flows into the stormwater detention ponds. It depicts the water level fluctuations and orifice settings for each pond. Notably, none of the three ponds experienced overflow, and the stream, indicated by the red line, also remained below overflow thresholds. For example, results related to pond3 are indicated in brown. The water level fluctuations of pond3 are represented by the thin brown solid line, while the control decision is depicted by the thicker brown solid line.

TABLE II: Overflow quantity comparison of different control syntheses.

Overflow $[m^3]$	Ponds	Stream
Centralized control	0.0	0.0
Compositional control	0.0	0.0
Decentralized control	6,019.2	0.0
Static (full-open)	0	12,690.0
Static (half-open)	3,826.2	9,442.8

Figure 8 presents qualitative experimental results of centralized control synthesis. The graphical analysis illustrates that both the ponds and the stream successfully avoided overflow events while maintaining elevated water levels in the ponds.

Figure 9 shows qualitative results of decentralized control synthesis. The stream remained free from overflow events, however, the ponds encountered prolonged and significant overflow incidents. In our model, the water overflowing from the ponds is redirected into the stream, thereby affecting the rise in the stream's water level. Each of the three ponds underwent independent controller synthesis, with a uniform application of the cost function used for learning across all ponds. This cost function considered both the individual water level of each pond and that of the stream.

Table II summarizes the quantitative results of the analysis in terms of overflow quantity (measured in m^3). It shows that both Centralized and Compositional control synthesis methods successfully prevented overflow. In contrast, the Decentralized control synthesis resulted in an overflow of 6,019.2 m^3 from the ponds. Further analysis, using a static controller as a baseline for our learning-based control synthesis, indicates that a fully open orifice setting led to a stream overflow of 12,690 m^3. Additionally, a half-open orifice resulted in overflow in both the ponds and the stream.

Table III provides a summary of the quantitative results from the analysis, focusing on the sedimentation rates. It is important to note that the results are dimensionless, as they represent relative rates rather than absolute quantities with units. In our experiments, the lower costs indicate better performance. The experimental results show that the centralized control yields the lowest cost, followed by the compositional controller. The low cost associated with decentralized control is attributed to the water level being at its maximum when overflow occurs, resulting in a minimum cost value of zero for that period. Furthermore, while sedimentation costs under conditions where no overflow occurs are comparable, it cannot be conclusively stated that the decentralized controller performs better than the centralized or compositional controllers. This is because, as observed in Table II, the occurrence of overflow undermines the apparent low sedimentation cost advantage of the decentralized approach.

In the field of learning-based control synthesis, which is known for its high computational demands, time efficiency stands out as a crucial evaluation metric [26]. Our current experiments are aimed at synthesizing controllers for ponds,

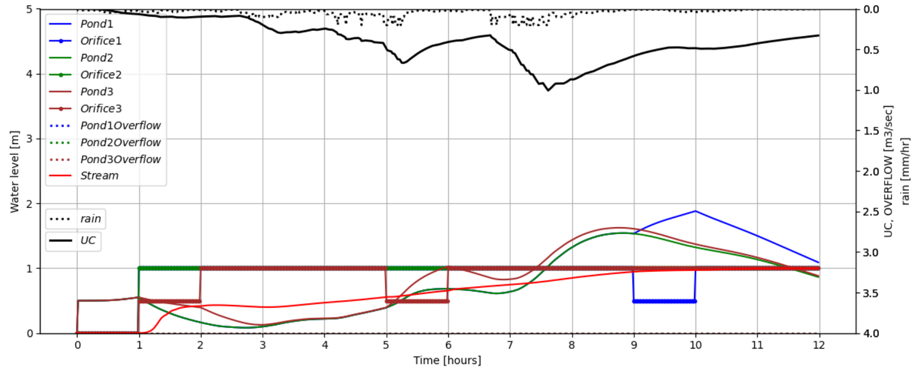

Fig. 7: Result of compositional synthesis control. There is no overflow from any of the ponds or the stream.

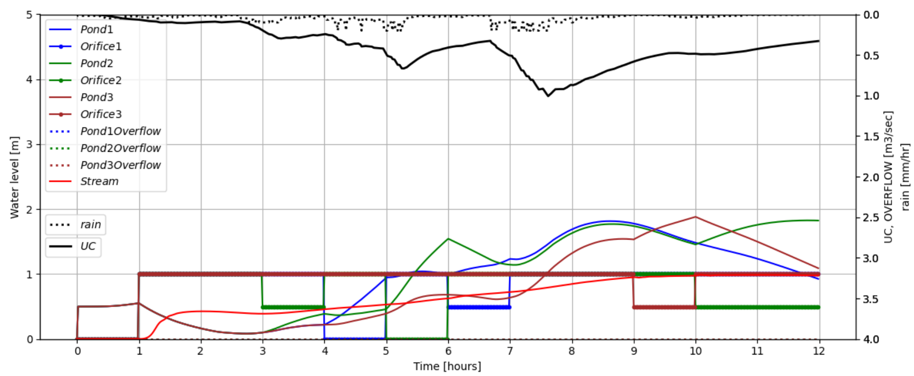

Fig. 8: Result of centralized synthesis control. There is no overflow from the stream nor any of the ponds.

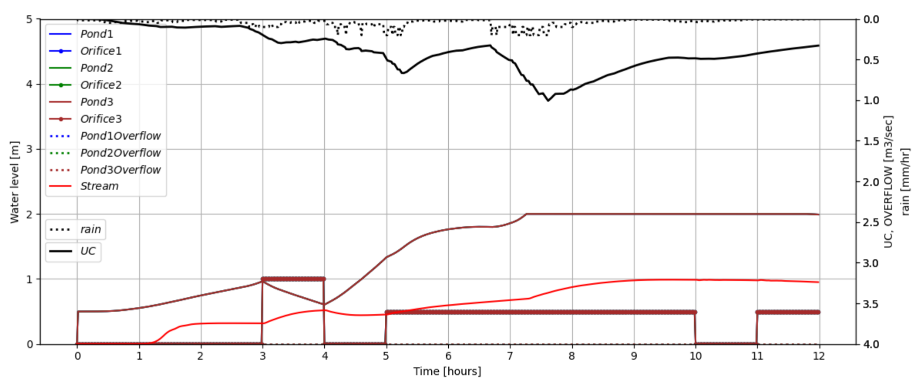

Fig. 9: Result of decentralized synthesis control. There is no overflow of the stream, but all three ponds experienced the same amount of overflow ($2,006.4m^3$).

TABLE III: Sedimentation rate of different control syntheses.

Sedimentation Cost [-]	Pond 1	Pond 2	Pond 3
Centralized control	388.62	358.56	426.97
Compositional control	426.97	458.15	425.74
Decentralized control	208.46	208.46	208.46

showing that although feasible, the scalability of centralized control synthesis presents significant challenges. As outlined in Table IV, the synthesis time for decentralized control is 379 min for a 1,000 runs [21], where the run refers to a configurable parameter that determines the maximum number of iterations in the learning process within UPPAAL STRATEGO. The compositional control synthesis consumes 503 minutes, whereas the centralized control synthesis demands 955 minutes, rendering it the most time-intensive approach among those under evaluation. With 3,000 and 10,000 runs, the computational time behavior for

TABLE IV: Time consumption comparison of different control syntheses.

Time consumption [min]	1,000 runs	3,000 runs	10,000 runs
Centralized control	955	1,897	5,450
Compositional control	503	1,002	2,985
Decentralized control	379	996	2,561

each control synthesis remains consistent compared to what we observe with 1,000 runs.

VII. Conclusion and Discussion

In this paper, we present a method for synthesizing compositional controllers for hybrid systems modeled using HMDP (Hybrid Markov Decision Processes). Our approach involves: (i) evaluating the impact of the system under both centralized and decentralized controllers, using the latter as a baseline for comparison; and (ii) analyzing the computational time associated with each controller type, thereby providing a clear comparative assessment. The compositional nature of our controller harmonizes the advantages of centralized proficiency with the computational merits of decentralized approaches. We observed potential stream overflows when multiple detention ponds discharged simultaneously in decentralized. Consequently, our model, emphasizing discharge timing and volume, becomes instrumental in averting such events. Furthermore, the application of reinforcement learning ensures the system's capability to sustain optimal water levels and reduce overflow risks. In subsequent studies, we plan to increase the number of ponds and expand the geographical scope to measure the continued efficiency of compositional control as the system becomes more complex. Through this, we could validate the scalability and adaptability of the framework we provided.

References

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th design automation conference*, 2010, pp. 731–736.

[2] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *Proceedings of 55th Conference on Decision and Control*. IEEE, 2016, pp. 5700–5707.

[3] A. Eqtami and A. Girard, "A quantitative approach on assume-guarantee contracts for safety of interconnected systems," in *Proceedings of 18th European Control Conference*. IEEE, 2019, pp. 536–541.

[4] G. Frehse, Z. Han, and B. Krogh, "Assume-guarantee reasoning for hybrid I/O-automata by over-approximation of continuous interaction," in *Proceedings of 43rd Conference on Decision and Control*, vol. 1. IEEE, 2004, pp. 479–484.

[5] L. Benvenuti, A. Ferrari, E. Mazzi, and A. S. Vincentelli, "Contract-based design for computation and verification of a closed-loop hybrid system," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2008, pp. 58–71.

[6] K. G. Larsen, M. Mikučionis, M. Muniz, J. Srba, and J. H. Taankvist, "Online and compositional learning of controllers with application to floor heating," in *Proceedings of 22nd International Conference Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2016, pp. 244–259.

[7] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *Proceedings of 55th Conference on Decision and Control*, 2016, pp. 5700–5707.

[8] M. Schütze, A. Campisano, H. Colas, W. Schilling, and P. A. Vanrolleghem, "Real time control of urban wastewater systems – where do we stand today?" *Journal of hydrology*, vol. 299, no. 3-4, pp. 335–348, 2004.

[9] W. D. Xu, M. J. Burns, F. Cherqui, and T. D. Fletcher, "Enhancing stormwater control measures using real-time control technology: a review," *Urban Water Journal*, vol. 18, no. 2, pp. 101–114, 2021.

[10] A. Mullapudi, M. J. Lewis, C. L. Gruden, and B. Kerkez, "Deep reinforcement learning for the real time control of stormwater systems," *Advances in water resources*, vol. 140, p. 103600, 2020.

[11] W. Tian, Z. Liao, G. Zhi, Z. Zhang, and X. Wang, "Combined sewer overflow and flooding mitigation through a reliable real-time control based on multi-reinforcement learning and model predictive control," *Water Resources Research*, vol. 58, no. 7, p. e2021WR030703, 2022.

[12] B. D. Bowes, C. Wang, M. B. Ercan, T. B. Culver, P. A. Beling, and J. L. Goodall, "Reinforcement learning-based real-time control of coastal urban stormwater systems to mitigate flooding and improve water quality," *Environmental Science: Water Research & Technology*, vol. 8, no. 10, pp. 2065–2086, 2022.

[13] K. Chen, H. Wang, B. Valverde-Pérez, S. Zhai, L. Vezzaro, and A. Wang, "Optimal control towards sustainable wastewater treatment plants based on multi-agent reinforcement learning," *Chemosphere*, vol. 279, p. 130498, 2021.

[14] C. S. de Witt, B. Peng, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson, "Deep multi-agent reinforcement learning for decentralized continuous cooperative control," *arXiv preprint arXiv:2003.06709*, vol. 19, 2020.

[15] M. A. Goorden, K. G. Larsen, J. E. Nielsen, T. D. Nielsen, M. R. Rasmussen, and J. Srba, "Learning safe and optimal control strategies for storm water detention ponds," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 13–18, 2021.

[16] M. Zhang, Z. Xu, Y. Wang, S. Zeng, and X. Dong, "Evaluation of uncertain signals' impact on deep reinforcement learning-based real-time control strategy of urban drainage systems," *Journal of Environmental Management*, vol. 324, p. 116448, 2022.

[17] A. L. Mollerup, P. S. Mikkelsen, D. Thornberg, and G. Sin, "Controlling sewer systems–a critical review based on systems in three eu cities," *Urban Water Journal*, vol. 14, no. 4, pp. 435–442, 2017.

[18] M. Eulogi, S. Ostojin, P. Skipworth, S. Kroll, J. Shucksmith, and A. Schellart, "Comparing methods to place adaptive local rtc actuators for spill volume reduction from multiple CSOs," *Journal of Hydroinformatics*, vol. 24, no. 1, pp. 78–92, 2022.

[19] P. Ashok, J. Křetínskỳ, K. G. Larsen, A. Le Coënt, J. H. Taankvist, and M. Weininger, "Sos: safe, optimal and small strategies for hybrid Markov decision processes," in *Proceedings of 16th International Conference on Quantitative Evaluation of Systems*. Springer, 2019, pp. 147–164.

[20] J. T. Mobley and T. B. Culver, "Design of outlet control structures for ecological detention ponds," *Journal of Water Resources Planning and Management*, vol. 140, no. 2, pp. 250–257, 2014.

[21] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist, "Uppaal stratego," in *Proceedings of 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2015, pp. 206–211.

[22] M. Jaeger, P. G. Jensen, K. G. Larsen, A. Legay, S. Sedwards, and J. H. Taankvist, "Teaching stratego to play ball: Optimal synthesis for continuous space MDPs," in *Proceedings of 17th International Symposium Automated Technology for Verification and Analysis*. Springer, 2019, pp. 81–97.

[23] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer, 2013.

[24] M. A. Goorden, P. G. Jensen, K. G. Larsen, M. Samusev, J. Srba, and G. Zhao, "STOMPC: Stochastic model-predictive control with Uppaal Stratego," in *Proceedings of the International Symposium on Automated Technology for Verification and Analysis*. Springer, 2022, pp. 327–333.

[25] B. E. McDonnell, K. Ratliff, M. E. Tryby, J. J. X. Wu, and A. Mullapudi, "PySWMM: The Python interface to stormwater management model (SWMM)," *Journal of Open Source Software*, vol. 5, no. 52, p. 2292, 2020.

[26] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," *Advances in neural information processing systems*, vol. 20, 2007.