Model Reduction of Linear Stochastic Systems with Preservation of sc-LTL Specifications

M.H.W. Engelaar[1], L. Romao[2], Y. Gao[2] , M. Lazar[1], A. Abate[2], and S. Haesaert[1]

Abstract— **We propose a correct-by-design controller synthesis framework for discrete-time linear stochastic systems that provides more flexibility to the overall abstraction framework of stochastic systems. Rather than directly abstracting the original dynamics, which can be large-scale and complex, we propose an intermediate step that leverages weak Gaussian realization theory and Kalman filtering techniques to obtain a related, discrete-time stochastic dynamical system that is simpler, and more prone to abstraction methods. We also propose a controller refinement algorithm and show correctness of the overall approach in enforcing synthetically co-safe Linear Temporal Logic properties. In general, the generated simplified stochastic dynamical systems are time-varying, but, under some technical conditions, will become time-invariant. We illustrate our theoretical findings with an example that supports the proposed correct-by-design framework and that illustrates how model reduction of stochastic models can be achieved.**

I. Introduction

Verifying or enforcing properties for dynamical systems entail dealing with inherent model complexity and modelling uncertainty. Engineering fields such as robotics [18], autonomous driving [4], aeronautics [17], astronautics [14], and energy regulation [7], [8], would benefit from the development of principled methods that could adequately tackle these two sources of modelling errors. A common framework to safeguard the system behaviour against model uncertainty is to treat it within a stochastic framework and use feedback, in which a probability distribution is assumed to govern the unwanted uncertainty, and one tries to perform controller design taking such an information into account.

The exact type of techniques that are employed to design the controller will depend on the specifications, and the control community has mostly focused on simpler temporal properties, such as stability, reachability, and safety (or forward invariance), which can be established using Lyapunov-type of arguments. More complex properties, such as reach-avoid and liveliness [3], and more generally those described by temporal logics such as Linear Temporal Logic (LTL), synthetically co-safe Linear Temporal Logic (sc-LTL) and Probabilistic Computation Tree Logic (PCTL), require a different set of tools to perform controller design. Abstractions of dynamical systems constitute an important framework for feedback controller design of these more complex properties. A discrete transition system (probabilistic or (non-)deterministic) is associated with the original dynamics, usually by partitioning the state space of the original dynamics and matching discrete states with elements of the partition, in a way that controllers designed using this simpler model lead to valid controllers for the original dynamics. Several computational tools are available (AMYTISS [12], FAUST^2 [16], StocHy [5] and SySCoRe [19]) that allow us to synthesise feedback controllers using abstraction framework.

Regrettably, due to the fact that the partition complexity grows exponentially with the dimension of the state space, the discretization step in these tools scale poorly, thus hampering their use in large-scale, safety-critical applications. In [8], [19], it was shown that model reduction techniques can mitigate the curse of dimensionality. However in [8], [19], the allowed model order reductions are limited to those that satisfy the simulation relation in [8]. In this paper, we alleviate the curse of dimensionality by developing a new correct-by-design technique for stochastic models which allows us to reduce the dimension of the original stochastic representation, while retaining essential features for the controller design process. This reduction moves beyond what can currently be quantified by simulation relations in [8], [10], [15].

More specifically, we rely on weak Gaussian stochastic realization theory [20] via Kalman filtering [1], [6] to remove redundant state information from the original system and pave the road to perform model reduction. First, we abstract away redundant information by creating a partial observable stochastic system [11]. Then, we use Kalman filtering to create a new model that is a weak stochastic realization of the former partial observable model. We show that this procedure leads to a correct-by-design framework and opens the road for discretization techniques on lower dimension representation of the dynamics rather than in the original, high-dimensional one. The controller refinement step is inspired by the results in [8].

In a nutshell, we use Kalman filtering to perform model reduction of the original stochastic dynamics. We create an abstraction for the original stochastic system and show how to refine the controller, obtained using the abstraction that is a reduced-order system, into a controller to the original, complex stochastic model, thus proving correctness of the proposed correct-by-design-framework for the considered class of stochastic systems.

The remainder of the paper is organized as follows. First, in Section II, some preliminary knowledge will be given. Next in Section III, both the problem statement and the approach will be introduced. Afterwards, in Section IV, both

[1] Department of Electrical Engineering (Control Systems Group), Eindhoven University of Technology, The Netherlands

[2] Department of Computer Science, University of Oxford, United Kingdom

Emails:{m.h.w.engelaar, m.lazar, s.haesaert}@tue.nl; {licio.romao, yulong.gao, alessandro.abate}@cs.ox.ac.uk

Manuscript 1263 submitted to 2023 62nd IEEE Conference on Decision and Control (CDC) . Received March 31, 2023.

the abstraction procedure and the controller refinement algorithm will be explained, yielding time-varying abstractions. Next in Section V, conditions for which the abstraction becomes time-invariant, are considered. Afterwards, in Section VI a case study will be given to illustrate the abstraction procedure and controller refinement algorithm, while also containing some additional analysis with regards to model order reduction and lack of completeness. Finally, in Section VII, a summary and conclusion is given.

II. Preliminaries

Notation. $\mathbb{N}_0$ denotes the set of all natural numbers including zero. $\mathbb{R}^n$ denotes the set of all n-dimensional real valued vectors. $\mathbb{X}^p := \prod_p \mathbb{X}$ denotes the p-time Cartesian product of $\mathbb{X}$. $X \succ 0$ denotes a matrix X that is strictly positive definite.

Controlled linear stochastic systems. We consider a linear time-invariant (LTI) stochastic system $\mathbf{M}$ given by

$$\mathbf{M} : \begin{cases} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ z(t) &= Hx(t), \end{cases} \tag{1}$$

where $x \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state, $u \in \mathbb{U} \subseteq \mathbb{R}^m$ is the input, $z \in \mathbb{Z} \subseteq \mathbb{R}^p$ is the (performance) output, $x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$ is the initial state and $w(t)$ is an independent, identically distributed noise disturbance with distribution $w(t) \sim \mathcal{N}(0, Q_w)$.

Finite executions of $\mathbf{M}$ are alternating sequences of states and inputs ending in a state, such as $\boldsymbol{\omega}_N^{\text{fin}} = x(0)u(0)x(1)u(1)\dots x(N-1)u(N-1)x(N)$, which satisfy equations (1) for some finite noise sequence $\boldsymbol{w} = w(0)w(1)w(2)\cdots, w(N-1)$, where $x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $w(k) \sim \mathcal{N}(0, Q_w)$ for all $k \in \{0, 1, \cdots, N-1\}$. We denote the set of all finite executions of length N by $\mathcal{E}_N \subseteq (\mathbb{X} \times \mathbb{U})^N \times \mathbb{X}$.

In its most general setting a controller $\mathbf{C}$ is a sequence of policies $\mathbf{C} := C(0)C(1)C(2)\cdots$, such that $C(k)$ is a map of the available history to the set of inputs $C(k) : \mathcal{E}_k \to \mathbb{U}$ for which the chosen control inputs are given by $u(k) = \mathbf{C}_k(\boldsymbol{\omega}_k^{\text{fin}})$, with $\mathbf{C}_k := C(k)$.

The controlled stochastic system $\mathbf{C} \times \mathbf{M}$ is obtained by composing $\mathbf{C}$ with $\mathbf{M}$. Due to the stochastic nature of $\mathbf{M}$, each evaluation of $\mathbf{C} \times \mathbf{M}$ will produce an output trajectory $\boldsymbol{z} = z(0)z(1)z(2)\cdots$ that is also stochastic. Equivalently, the output $\boldsymbol{z}$ can be interpreted as a realization of the probability distribution described by the autonomous system $\mathbf{C} \times \mathbf{M}$, written as $\boldsymbol{z} \sim \mathbf{C} \times \mathbf{M}$.

Linear Temporal Logic. Consider a set $AP = \{p_1, \dots, p_n\}$ of atomic propositions and the corresponding alphabet $\Sigma := 2^{AP}$ with letters $\pi \in 2^{AP}$ composed by all subsets of AP. Let L be a labelling map that maps outputs z to letters in the alphabet, that is, $L : \mathbb{Z} \to \Sigma$. Then we have that any output trajectory $\boldsymbol{z}$ defines an infinite string of letters $\boldsymbol{\pi} = \pi_0\pi_1\pi_2\dots$ with $\pi_i = L(z_i)$. We refer to these infinite strings of letters $\boldsymbol{\pi} \in \Sigma^\infty$ as words.

We now introduce LTL formulas that can be evaluated over these words. For this paper, we assume without loss of generality that the formulas are given as syntactically co-safe Linear-time Temporal Logic specifications (sc-LTL). The syntax of sc-LTL is given by

$$\phi ::= \top \mid p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2.$$

Let a suffix of $\boldsymbol{\pi}$ be denoted as $\boldsymbol{\pi}_k := \pi_k\pi_{k+1}\pi_{k+2}\dots$, then the semantics of sc-LTL are given as follows:

true	$\boldsymbol{\pi}_k \vDash \top$	$\iff$ true
atomic prop.	$\boldsymbol{\pi}_k \vDash p$	$\iff p \in \pi_k$
negation	$\boldsymbol{\pi}_k \vDash \neg p$	$\iff p \notin \pi_k$
and	$\boldsymbol{\pi}_k \vDash \phi_1 \wedge \phi_2$	$\iff \boldsymbol{\pi}_k \vDash \phi_1$ and $\boldsymbol{\pi}_k \vDash \phi_2$
or	$\boldsymbol{\pi}_k \vDash \phi_1 \vee \phi_2$	$\iff \boldsymbol{\pi}_k \vDash \phi_1$ or $\boldsymbol{\pi}_k \vDash \phi_2$
next	$\boldsymbol{\pi}_k \vDash \bigcirc\phi$	$\iff \boldsymbol{\pi}_{k+1} \vDash \phi$
until	$\boldsymbol{\pi}_k \vDash \phi_1 \mathcal{U} \phi_2$	$\iff \exists i \in \mathbb{N}_0 : \boldsymbol{\pi}_{k+i} \vDash \phi_2$ and $\forall j \in \{0, 1, \dots i-1\}$: $\boldsymbol{\pi}_{k+j} \vDash \phi_1$.

If $\boldsymbol{\pi}_0 \vDash \phi$, then the word $\boldsymbol{\pi}$ satisfies the specification. Extensions to more complex operators can be made. For instance, the *eventual* operator $\Diamond$ can be defined as $\Diamond\phi := \top\mathcal{U}\phi$. More details can be found in [2], [3].

Note that in the sequel, we will use $\mathbf{z} \vDash \phi$ to denote $\boldsymbol{\pi} \vDash \phi$ with $\boldsymbol{\pi}$ defined by labelling $\mathbf{z}$ with L.

III. Problem Statement

Stochastic Correct-by-Design Control Synthesis. Let us consider the goal of designing a controller $\mathbf{C}$ that ensures that the outputs of the controlled system $\mathbf{C} \times \mathbf{M}$ satisfy a given specification ϕ. More precisely, given the stochastic nature of $\mathbf{M}$, it is natural to require that the specification ϕ is satisfied by the controlled system $\mathbf{C} \times \mathbf{M}$ with probability at least p. Let us denote this satisfaction probability as $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi)$, i.e., $\mathbb{P}(\mathbf{z} \vDash \phi \mid \mathbf{z} \sim \mathbf{C} \times \mathbf{M})$. Then, the objective is to synthesize $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$. We refer to this as *stochastic correct-by-design control synthesis*.

This type of control synthesis for stochastic systems is an active area of research [13] and multiple tools exist that can tackle the control for (sc-)LTL specifications, including but not limited to AMYTISS [12], FAUST^2 [16], StocHy [5], and SySCoRe [19]. However, as mentioned in the introduction, these tools scale poorly with state space and disturbance dimensions.

Problem statement. To mitigate scaling problems in stochastic correct-by-design control synthesis, we are interested in designing an abstract model for which the stochastic control synthesis problem is substantially simpler, while still preserving correctness with respect to sc-LTL specifications.

More precisely, our goal is to construct a reduced-order abstract model $\bar{\mathbf{M}}$ such that for any correct-by-design controller $\bar{\mathbf{C}}$ synthesized for the abstract model, a correct-by-design controller $\mathbf{C}$ can be obtained for the original model $\mathbf{M}$ with equal satisfaction probability p, i.e.,

$$\forall \bar{\mathbf{C}} : \mathbb{P}(\bar{\mathbf{C}} \times \bar{\mathbf{M}} \vDash \phi) \geq p,\ \exists \mathbf{C} : \mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p. \tag{2}$$

In this paper, we specifically look for a constructive approach to this problem.

Literature. To reduce the complexity of stochastic systems, model order reduction based methods have been developed in the past; see, e.g., [15] and references therein. Instances of this include, amongst others, [8], [10], [15], which introduced (approximate) stochastic simulation relations, stochastic simulation functions, and stochastic bisimulation relations. These notions allowed higher-order general Markov Decision processes, continuous-time stochastic hybrid systems, and linear stochastic systems to be simulated by lower-order systems of the same type, all while retaining (approximate) equivalency with regard to their output (distribution).

Approach. Our approach, on the other hand, will hinge on weak Gaussian stochastic realization theory [20] and leverages optimal Kalman filtering [1], [6] to obtain reduced-order abstract models. We will show that different levels of abstractions can be created, and in some cases, we can reduce the dimension further than previous model order reduction-based methods allowed. Inspired by [8], we also derive how the satisfaction probability of specifications can be preserved via control refinement strategy. In the next sections, we will first consider the case in which the abstraction becomes time-varying, to afterwards consider the time-invariant case.

IV. Abstraction and Control Refinement: Time-varying Abstraction

Consider a linear stochastic system $\mathbf{M}$ as given by (1). To solve the stochastic correct-by-design control synthesis problem in (2), we introduce the abstraction procedure illustrated in Fig. 1 and the controller refinement algorithm represented in Algorithm 1.

The abstraction procedure hinges on the idea of removing state information from the original model $\mathbf{M}$ without influencing the performance output z. This is achieved in two steps: The first step removes potentially redundant information from $\mathbf{M}$ by introducing an observation output $y(t) = Cx(t)$. The result is a new stochastic system $\mathbf{M}_{\text{Obs}}$ that is partially observable[1].

The second step replaces the partially observable model by a fully observable equivalent model via optimal Kalman filtering. We refer to this model as the abstract model $\bar{\mathbf{M}}$.

With the abstraction procedure outlined, all that remains is to formally express the abstract model and prove that the procedure is valid. Hence, in the following subsections, a formal definition of the abstract model will be given. Next, it is shown that the original model $\mathbf{M}$ and the abstraction $\bar{\mathbf{M}}$ have the same performance output distribution whenever $u(t) = \bar{u}(t)$. Afterwards, a constructive proof will be given, in which a correct-by-design controller $\mathbf{C}$ will be obtained from any correct-by-design controller $\bar{\mathbf{C}}$, giving us the controller refinement algorithm. In this section, $\bar{\mathbf{M}}$ is a time varying model, but in the next section we present conditions under which a time-invariant abstraction can be obtained.

[1] The system $\mathbf{M}_{\text{Obs}}$ is a partially observable Markov decision process [11].

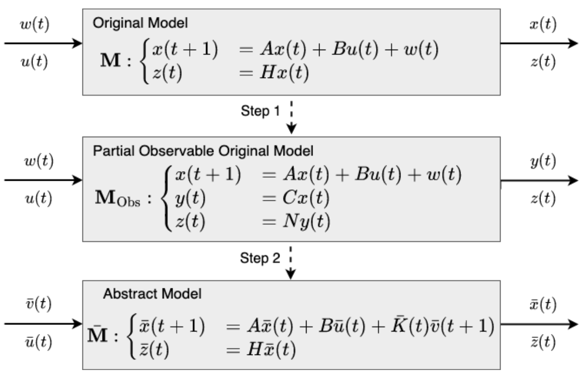
$w(t)$
$u(t)$
Original Model
$\mathbf{M}: \begin{cases} x(t+1) &= Ax(t)+Bu(t)+w(t) \\ z(t) &= Hx(t) \end{cases}$
$x(t)$
$z(t)$
Step 1
$w(t)$
$u(t)$
Partial Observable Original Model
$\mathbf{M}_{\text{Obs}}: \begin{cases} x(t+1) &= Ax(t)+Bu(t)+w(t) \\ y(t) &= Cx(t) \\ z(t) &= Ny(t) \end{cases}$
$y(t)$
$z(t)$
Step 2
$\bar{v}(t)$
$\bar{u}(t)$
Abstract Model
$\bar{\mathbf{M}}: \begin{cases} \bar{x}(t+1) &= A\bar{x}(t)+B\bar{u}(t)+\bar{K}(t)\bar{v}(t+1) \\ \bar{z}(t) &= H\bar{x}(t) \end{cases}$
$\bar{x}(t)$
$\bar{z}(t)$

Fig. 1. A block diagram illustrating the abstraction procedure. The distribution of $\bar{v}(t)$ depends on the distribution of $y(t)$. $\bar{K}(t)$ is the time-varying Kalman gain obtained from the Kalman filter equations.

A. Abstract Model Construction

Consider a linear stochastic system $\mathbf{M}$ as given by (1). Following the steps of the abstraction procedure Fig. 1, we now construct the abstract $\bar{\mathbf{M}}$ and the corresponding system matrices.

As a first step, we choose a matrix pair C and N with $C \in \mathbb{R}^{q \times n}$ and $N \in \mathbb{R}^{p \times q}$, such that $NC = H$ and $q < n$. We then construct the partial observable original model given by

$$\mathbf{M}_{\text{Obs}}: \begin{cases} x(t+1) &= Ax(t)+Bu(t)+w(t) \\ y(t) &= Cx(t) \\ z(t) &= Ny(t), \end{cases} \tag{3}$$

where $x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$, $y(t) \in \mathbb{R}^q$ is the observation output, and $w(t) \sim \mathcal{N}(0, Q_w)$.

For the next step, we use Kalman filtering techniques to estimate the states of (3). To this end, we denote by $x_K(t|t)$ the conditional expectation of $x(t)$, given the past information, that is, $x_K(t|t) = \mathbb{E}(x(t)|y(0)u(0)...u(t-1)y(t))$. We also denote by $P(t|t)$ the conditional variance of $x(t)$ under the same information, i.e., $P(t|t) = \text{Var}([x(t) - x_K(t|t)]^2| \cdots y(t))$. Quantities $x_K(t|t)$ and $P(t|t)$ are called, respectively, *a posteriori state estimate mean* and *a posteriori state variance*. The term a posteriori is used due to the fact that we use the most updated information, namely $y(t)$, in the definition of $x_K(t|t)$ and $P(t|t)$. The a priori information is defined similarly using the information $y(0)u(0)...u(t-1)$. A standard result in Kalman filtering theory is the optimal mean-square filter which leads to the following updates on the quantities, see also [1], [6].

$$\begin{aligned} x_K(t|t) &= x_K(t|t-1) \\ &\quad + K(t)[y(t) - Cx_K(t|t-1)] \end{aligned} \tag{4a}$$

$$x_K(t+1|t) = Ax_K(t|t) + Bu(t) \tag{4b}$$

$$P(t|t) = P(t|t-1) - K(t)CP(t|t-1) \tag{4c}$$

$$P(t+1|t) = AP(t|t)A^T + Q_w \tag{4d}$$

$$K(t) = \ P(t|t-1)C^T[CP(t|t-1)C^T]^{-1} \tag{4e}$$

$$x_K(0|-1) = \mu_0 \text{ and } P(0|-1) = \Sigma_0. \tag{4f}$$

Note that as there is no noise term on the output y in model (3), the Kalman filter equations reduce to those given above.

It is well known that the error between the a priori prediction of the next state $x_K(t|t-1)$ and the measurement $y(t)$ defines a white noise sequence. As is common, we refer to this Gaussian white noise sequence as the innovation and define it as

$$v(t) = y(t) - Cx_K(t|t-1). \tag{5}$$

The mean and covariance of v can be computed directly, giving us $\mu_v(t) = 0$ and $\Sigma_v(t) = CP(t|t-1)C^T$, see also [20, Theorem 14.4.2].

We can now leverage the innovation sequence to define a new linear stochastic system that is equivalent to the original system (see also [20]). This new system, referred to as the innovation process, builds on the equations of the a priori state estimation equation of $x_K(t+1|t)$ and is given as

$$\hat{\mathbf{M}} : \begin{cases} \hat{x}(t+1) & = A\hat{x}(t) + B\hat{u}(t) + \hat{K}(t)\hat{v}(t), \\ \hat{y}(t) & = C\hat{x}(t) + \hat{v}(t) \\ \hat{z}(t) & = N\hat{y}(t) \end{cases} \tag{6}$$

where

$$\begin{aligned} \hat{x}(0) &= \mu_0,\ \hat{v}(t) \sim \mathcal{N}(0, Q_{\hat{v}}(t)), \\ \hat{K}(t) &= A\hat{P}(t)C^T\left(C\hat{P}(t)C^T\right)^{-1}, \\ \hat{P}(t+1) &= A\hat{P}(t)A^T + Q_w - \hat{K}(t)C\hat{P}(t)A^T, \\ Q_{\hat{v}}(t) &= C\hat{P}(t)C^T, \text{ and } \hat{P}(0) = \Sigma_0. \end{aligned}$$

Although, this is the most commonly used version, we will now define an alternative based on the a posteriori state estimate equations. We refer to this as the *current* innovation process given by

$$\bar{\mathbf{M}} : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + \bar{K}(t)\bar{v}(t+1) \\ \bar{y}(t) & = CA\bar{x}(t-1) + CB\bar{u}(t-1) + \bar{v}(t) \\ \bar{z}(t) & = N\bar{y}(t) \end{cases} \tag{7}$$

where

$$\begin{aligned} \bar{x}(0) &\sim\ \mathcal{N}(\mu_0, \Sigma_0 - \bar{P}(0)), && \text{(8a)} \\ \bar{v}(t) &\sim \mathcal{N}(0, Q_{\bar{v}}(t)), \quad Q_{\bar{v}}(t+1) = C\bar{R}(t)C^T && \text{(8b)} \\ \bar{K}(t) &= \bar{R}(t)C^T(C\bar{R}(t)C^T)^{-1}, && \text{(8c)} \\ \bar{P}(t+1) &= \bar{R}(t) - \bar{K}(t)C\bar{R}(t), && \text{(8d)} \\ \bar{R}(t) &= A\bar{P}(t)A^T + Q_w, && \text{(8e)} \\ \bar{P}(0) &= \Sigma_0 - \Sigma_0 C^T[C\Sigma_0 C^T]^{-1}C\Sigma_0. && \text{(8f)} \end{aligned}$$

In the appendix additional information is given with regards to the derivation of the innovation processes.

In the remainder, we will make the following assumption.

Assumption 1: Assume that $C\Sigma_0 C^T \succ 0$ and $CQ_wC^T \succ 0$.

Notice that $\bar{y}(t) = C\bar{x}(t) - C\bar{K}(t-1)\bar{v}(t) + \bar{v}(t)$. Since $\bar{K}(t) = \bar{R}(t)C^T(C\bar{R}(t)C^T)^{-1}$, we have that $CK(t-1) = I$ and hence that $C\bar{x}(t) + (I - C\bar{K}(t-1))\bar{v}(t) = C\bar{x}(t)$. Therefore we get the following abstract model

$$\bar{\mathbf{M}} : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + \bar{K}(t)\bar{v}(t+1) \\ \bar{z}(t) & = H\bar{x}(t). \end{cases} \tag{9}$$

with initial state $\bar{x}(0)$ given by (8a), innovation $\bar{v}(t)$ given by (8b) and $\bar{K}$ as computed in (8c). This model is a *weak stochastic realization* of $\mathbf{M}_{\text{Obs}}$ in (3). Note that it is not a realization of the original model $\mathbf{M}$.

We can now prove that the original model $\mathbf{M}$ and the abstract model $\bar{\mathbf{M}}$ have equivalent performance outputs for any given sequence of inputs for which $u(t) = \bar{u}(t), \forall t \in \mathbb{N}_0$.

Theorem 1: $\mathbf{M}$ and $\bar{\mathbf{M}}$ have the same performance output distributions $\boldsymbol{z}$ and $\bar{\boldsymbol{z}}$ if $u(t) = \bar{u}(t)$, $\forall t \in \mathbb{N}_0$.

Proof: The proof can be found in the appendix. □

B. Controller Refinement

What remains to be shown, is that for any correct-by-design controller $\bar{\mathbf{C}}$ designed for the abstract model $\bar{\mathbf{M}}$, there exists a correct-by-design controller $\mathbf{C}$ for the original model $\mathbf{M}$. We use the auxiliary model $\mathbf{M}_{\text{Obs}}$ to define a control refinement algorithm next. This algorithm is presented in Algorithm 1.

Algorithm 1 Obtain controller $\mathbf{C}$ based on controller $\bar{\mathbf{C}}$

```
Given: $\mathbf{M}$, $\bar{\mathbf{M}}$, $\bar{\mathbf{C}}$
set $t := 0$ and compute $K(0) := \Sigma_0 C^T(C\Sigma_0 C^T)^{-1}$,
draw $x(0)$ from $\mathcal{N}(\mu_0, \Sigma_0)$,
compute $\bar{x}(0) = \mu_0 + K(0)(Cx(0) - C\mu_0)$,
loop
    obtain $\bar{u}(t)$ according to $\bar{\mathbf{C}}$,
    set $u(t) = \bar{u}(t)$, { ← Implementing $\mathbf{C}$}
    draw $x(t+1)$ from $\mathbf{M}$ and get $y(t+1) = Cx(t+1)$,
    compute $\bar{v}(t+1) = y(t+1) - CA\bar{x}(t) - CB\bar{u}(t)$,
    compute $\bar{x}(t+1) = A\bar{x}(t) + B\bar{u}(t) + \bar{K}(t)\bar{v}(t+1)$
    take $t = t + 1$.
end loop
```

An illustrative description of Algorithm 1 is given in Fig. 2, where the combination of all blocks excluding the $\mathbf{M}$ block, define the controller $\mathbf{C}$.

The next theorem constitutes one of the main contributions of this paper, as it provides a solution to the problem description in equation (2).

Theorem 2: Under Assumption 1, let ϕ be any sc-LTL specification and $p \in [0, 1]$. For all controllers $\bar{\mathbf{C}}$ such that $\mathbb{P}(\bar{\mathbf{C}} \times \bar{\mathbf{M}} \vDash \phi) \geq p$, there exists a controller $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$.

Proof: Consider $\mathbf{C}$ constructed based on Algorithm 1. Using standard Kalman filtering arguments, one can show that, at every time step, the realization of $\bar{v}(t+1)$ in Algorithm 1 is uncorrelated with the current state $\bar{x}(t+1)$ and has distribution given by (8b). Therefore, the embedded model $\bar{\mathbf{M}}$ has the same distribution as $\bar{\mathbf{M}}$ in (9) and thus the output of the embedded $\bar{\mathbf{M}}$ is equal to that of $\mathbf{M}$. Consequently, since $\mathbb{P}(\bar{\mathbf{C}} \times \bar{\mathbf{M}} \vDash \phi) \geq p$, we have that

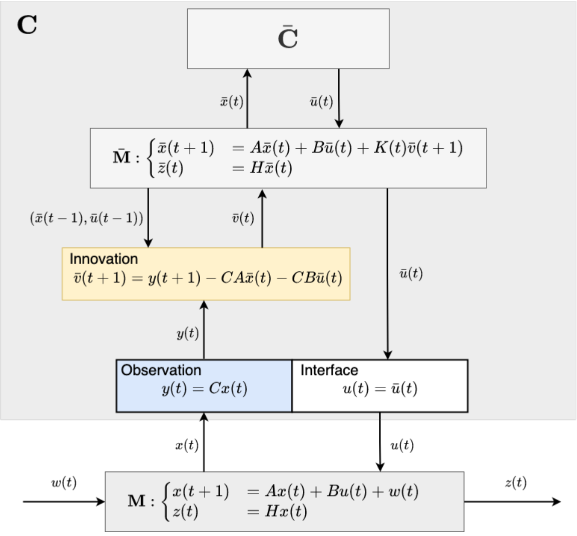
C
$\bar{\mathbf{C}}$
$\bar{x}(t)$
$\bar{u}(t)$
$\bar{\mathbf{M}} : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + K(t)\bar{v}(t+1) \\ \bar{z}(t) & = H\bar{x}(t) \end{cases}$
$(\bar{x}(t-1), \bar{u}(t-1))$
$\bar{v}(t)$
Innovation
$\bar{v}(t+1) = y(t+1) - CA\bar{x}(t) - CB\bar{u}(t)$
$\bar{u}(t)$
$y(t)$
Observation
$y(t) = Cx(t)$
Interface
$u(t) = \bar{u}(t)$
$x(t)$
$u(t)$
$w(t)$
$\mathbf{M} : \begin{cases} x(t+1) & = Ax(t) + Bu(t) + w(t) \\ z(t) & = Hx(t) \end{cases}$
$z(t)$

Fig. 2. A block diagram illustrating Algorithm 1.

$\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$, thus concluding the proof of the theorem. □

Notice that despite $\mathbf{M}$ and $\bar{\mathbf{M}}$ having the same state space dimension, the stochastic processes affecting these two systems have now different dimensions. More precisely, the noise input $w(t) \sim \mathcal{N}(0, Q_w)$ with $w(t) \in \mathbb{R}^n$ has now been replaced with the input $\bar{K}(t)v(t+1)$ with $v(t+1) \in \mathbb{R}^q$, where $q < n$. As of now, this reduction in complexity comes at the cost of having a time-varying system. In the next section, however, we will show that the results can be extended to obtain a time-invariant abstract system.

V. Abstraction and Control Refinement: Time-invariant Case

In this section, we investigate under which conditions we can obtain a time-invariant abstract model based on the procedure given in Section IV-A. More precisely, as in Section IV-A, we will use the definition of a partially observable model in step 1 to obtain an abstract model in step 2. In this case, we are looking for a time-invariant abstract model as depicted in Fig. 3.

First, we do this by placing a strict requirement on the initial distribution of the original system $x(0) \sim (\mu_0, \Sigma_0)$. Afterwards, we show how this condition can be relaxed.

A. Abstract Model Construction

Consider the discrete-time algebraic Riccati equation (DARE) adapted to the partially observable model (3), which does not have measurement noise, given by

$$X = AXA^T - AXC^T[CXC^T]^{-1}CXA^T + Q_w. \qquad (10)$$

We say that $X \succ 0$ is a stabilizing solution if $A - FC$ is stable for $F = AXC^T(CXC^T)^{-1}$. For more information on discrete-time algebraic Ricatti equations, see [9].

Consider the following lemma, for which the proof follows directly from the Kalman filter equations (4) and the abstraction procedure explained in section IV-A.

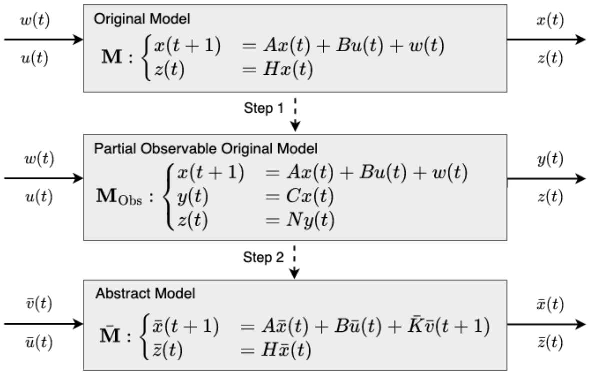
$w(t)$
$u(t)$
Original Model
$\mathbf{M} : \begin{cases} x(t+1) & = Ax(t) + Bu(t) + w(t) \\ z(t) & = Hx(t) \end{cases}$
$x(t)$
$z(t)$
Step 1
$w(t)$
$u(t)$
Partial Observable Original Model
$\mathbf{M}_{\text{Obs}} : \begin{cases} x(t+1) & = Ax(t) + Bu(t) + w(t) \\ y(t) & = Cx(t) \\ z(t) & = Ny(t) \end{cases}$
$y(t)$
$z(t)$
Step 2
$\bar{v}(t)$
$\bar{u}(t)$
Abstract Model
$\bar{\mathbf{M}} : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + \bar{K}\bar{v}(t+1) \\ \bar{z}(t) & = H\bar{x}(t) \end{cases}$
$\bar{x}(t)$
$\bar{z}(t)$

Fig. 3. A block diagram illustrating the abstraction procedure. The distribution of $\bar{v}(t)$ depends on the distribution of $y(t)$. $\bar{K}$ is the time-invariant Kalman gain obtained from the Kalman filter equations.

Lemma 3: Assume that $C\Sigma_0 C^T \succ 0$. If $\Sigma_0 = X \succ 0$ is a stabilizing solution X to the DARE (10), then the abstract model $\bar{\mathbf{M}}$ obtained from $\mathbf{M}$ is static, given by

$$\bar{\mathbf{M}} : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + \bar{K}\bar{v}(t+1) \\ \bar{z}(t) & = H\bar{x}(t) \end{cases} \qquad (11)$$

where $\bar{x}(0) \sim \mathcal{N}(\mu_0, X - \bar{P})$, with

$$\bar{v}(t) \sim \mathcal{N}(0, Q_{\bar{v}}), Q_{\bar{v}} = CXC^T,$$
$$\bar{K} = XC^T[CXC^T]^{-1}, \bar{P} = X - \bar{K}CX.$$

The main advantage of Lemma 3 is the guarantee of the current innovation process (11) being time-invariant, contrary to the current innovation process (9), which may or may not be time-invariant. Regrettably, most real-life engineering systems do not yield the result of Lemma 3, as, in general, Σ_0 will not solve the DARE (10). To alleviate this issue of practicality, we will consider an alternative version of Theorem 1 that assumes $\Sigma_0 - X \succ 0$.

Let $X \succ 0$ be the stabilizing solution to the DARE (10) and consider the abstract model $\bar{\mathbf{M}}^*$ obtained from $\mathbf{M}$, given by

$$\bar{\mathbf{M}}^* : \begin{cases} \bar{x}(t+1) & = A\bar{x}(t) + B\bar{u}(t) + \bar{K}\bar{v}(t+1) \\ \bar{z}(t) & = H\bar{x}(t) \end{cases} \qquad (12)$$

where $\bar{x}(0) \sim \mathcal{N}(\mu_0, \Sigma_0 - \bar{P})$, with

$$\bar{v}(t) \sim \mathcal{N}(0, Q_{\bar{v}}) \quad Q_{\bar{v}} = CXC^T$$
$$\bar{K} = XC^T[CXC^T]^{-1}, \bar{P} = X - \bar{K}CX.$$

Note that $\bar{\mathbf{M}}^*$ differs from the abstract model in (11) only in the initial distribution.

Theorem 4: Let $\mathbf{M}$ be given by (1) with initial state distribution $x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$, and let $X \succ 0$ be a stabilizing solution of the DARE (10). If $\Sigma_0 - X \succ 0$ and $CXC^T \succ 0$, then the abstract model $\bar{\mathbf{M}}^*$ given in (12) is such that $\mathbf{M}$ and $\bar{\mathbf{M}}^*$ have the same performance output distributions $\boldsymbol{z}$ and $\bar{\boldsymbol{z}}$ if $u(t) = \bar{u}(t)$, $\forall t \in \mathbb{N}_0$.

Proof: We will first show that the dynamics of $\bar{\mathbf{M}}^*$ are equal to the current innovation dynamics as derived in Section IV-A after doing two measurements steps. We take an additional measurement at $t = 0$ given by $\tilde{y}(0) = x(0) + \tilde{w}$, where $\tilde{w} \sim \mathcal{N}(0, R)$ with $R = (X^{-1} - \Sigma_0^{-1})^{-1}$. According to the Kalman filter equations [1], [6], the first measurement update step of the state estimate and covariance estimate are given by $\bar{x}_K(0|0) = \mu_0 + \Sigma_0[\Sigma_0 + R]^{-1}(\tilde{y}(0) - \mu_0) = \bar{\mu}_0$ and $\bar{P}_0 = \Sigma_0 - \Sigma_0[\Sigma_0 + R]^{-1}\Sigma_0$. Here the additional output noise $\tilde{w}$ had to be taken into account. Using the Woodbury matrix identity, we can subsequently derive the following equalities to show that $\bar{P}_0 = X$

$$\begin{aligned}
R &= (X^{-1} - \Sigma_0^{-1})^{-1} \\
R &= (\Sigma_0^{-1} - \Sigma_0^{-1} X \Sigma_0^{-1})^{-1} - \Sigma_0 \\
(R + \Sigma_0)^{-1} &= \Sigma_0^{-1} - \Sigma_0^{-1} X \Sigma_0^{-1} \\
\Sigma_0 (R + \Sigma_0)^{-1} \Sigma_0 &= \Sigma_0 - X \\
X &= \Sigma_0 - \Sigma_0 (R + \Sigma_0)^{-1} \Sigma_0.
\end{aligned}$$

As before, after applying the second measurement $y(0) = Cx(0)$, we regain the current state estimate $\bar{x}(0) = \bar{\mu}_0 + \bar{K}(Cx(0) - C\bar{\mu}_0)$ and the variance of the current state estimate $\bar{P}$. The effect of both measurements updates is captured by $\bar{x}(0) \sim \mathcal{N}(\mu_0, \Sigma_0 - \bar{P})$. To complete the proof, we can now continue using the abstraction procedure as detailed in section IV-A and the proof of Theorem 1. This would be equivalent to applying the abstraction procedure to a system $\mathbf{M}$ with initial distribution given by $x(0) \sim \mathcal{N}(\bar{\mu}_0, X)$. Since X solves the DARE (10), the result follows directly from Lemma 3 and $\bar{\mu}_0 \sim \mathcal{N}(\mu_0, \Sigma_0 - X)$, thereby finishing the proof of the theorem. □

While the procedure to obtaining the modified current innovation process (12) is equivalent to that explained in Section IV-A, a slight modification is made by assuming an additional output measurement is available at $t = 0$, thereby obtaining a modified version of the current innovation process compared to current innovation processes (9) and (11).

B. Controller Refinement Algorithm

Should the hypothesis of Lemma 3 hold true, Algorithm 1 can still be applied and Theorem 2 can be rephrased as follows.

Corollary 5: Assume that $\Sigma_0 \succ 0$ is a stabilizing solution to the DARE (10), and $C\Sigma_0 C^T \succ 0$, and let ϕ be any sc-LTL specification and $p \in [0, 1]$. For all $\bar{\mathbf{C}}$ such that $\mathbb{P}(\bar{\mathbf{C}} \times \bar{\mathbf{M}} \vDash \phi) \geq p$, there exists a controller $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$.

For the more relaxed condition on the initial distribution of $\mathbf{M}$, with abstract model $\bar{\mathbf{M}}^*$ (12), Algorithm 1 needs to be slightly modified resulting in Algorithm 2. Note that as in the proof of Theorem 4, the algorithm now uses an auxiliary step to ensure that the initialization is resolved correctly.

Based on Algorithm 2, we can now extend the result in Theorem 2, to the general time-invariant case.

Theorem 6: Let $X \succ 0$ be a stabilizing solution of the DARE (10), and assume that $\Sigma_0 - X \succ 0$ and $CXC^T \succ 0$. Let ϕ be any sc-LTL specification and $p \in [0, 1]$. For all $\bar{\mathbf{C}}^*$ such that $\mathbb{P}(\bar{\mathbf{C}}^* \times \bar{\mathbf{M}}^* \vDash \phi) \geq p$, there exists a controller $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$.

Algorithm 2 Obtain controller $\mathbf{C}$ based on controller $\bar{\mathbf{C}}^*$.

```
Given: $\mathbf{M}$, $\bar{\mathbf{M}}^*$, $\bar{\mathbf{C}}^*$
set $t := 0$ and compute $L = \Sigma_0[\Sigma_0 + R]^{-1}$,
draw $x(0)$ from $\mathcal{N}(\mu_0, \Sigma_0)$ and draw $\tilde{w}$ from $\mathcal{N}(0, R)$,
compute $\bar{\mu}_0 = \mu_0 + L(x(0) + \tilde{w} - \mu_0)$,
compute $\bar{x}(0) = \bar{\mu}_0 + \bar{K}(Cx(0) - C\bar{\mu}_0)$,
loop
    obtain $\bar{u}(t)$ according to $\bar{\mathbf{C}}^*$,
    set $u(t) = \bar{u}(t)$, { $\leftarrow$ Implementing $\mathbf{C}$}
    draw $x(t+1)$ from $\mathbf{M}$ and get $y(t+1) = Cx(t+1)$,
    compute $\bar{v}(t+1) = y(t+1) - CA\bar{x}(t) - CB\bar{u}(t)$,
    compute $\bar{x}(t+1) = A\bar{x}(t) + B\bar{u}(t) + \bar{K}\bar{v}(t+1)$
    take $t = t + 1$.
end loop
```

Proof: The proof of the theorem follows directly from Algorithm 2, similar to Theorem. 2 with the minor additional steps based on Theorem 4. □

Due to loss of state information when constructing the current innovation process, in general, Theorem 2, Cor. 5 and Theorem 6 do not hold true when reversing the statement, that is, existence of $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq p$ does not imply existence of $\bar{\mathbf{C}}$ such that $\mathbb{P}(\bar{\mathbf{C}} \times \bar{\mathbf{M}} \vDash \phi) \geq p$. This will be further illustrated in the following section.

VI. Stochastic Correct-by-Design Control Synthesis

In this section, we will consider a case study to illustrate the abstraction procedure and the controller refinement algorithm. The same case study will also be used to show that under the right conditions, model order reduction can be achieved. We will end this section by illustrating that by reducing observable data, the abstract model will be limited, compared to the original model, with regards to synthesis of correct-by-design controllers.

Example 1: Consider the discrete-time stochastic system

$$\mathbf{M} : \begin{cases} x(t+1) &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) + w(t) \\ z(t) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t), \end{cases} \tag{13}$$

where $x(0) \sim \mathcal{N}(0, \Sigma_0)$ and $w \sim \mathcal{N}(0, Q_w)$ with

$$\Sigma_0 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \quad Q_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}.$$

Let $\phi := \square_{[1,100]} q$, where $q = L(z)$ if and only if $z \in [-1, 1]$, be the temporal specification, where $\square_{[i,j]} q$ means that q is always satisfied within the interval $[i, j]$. Our goal is to design a controller $\mathbf{C}$ such that $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq 0.95$.

Let us construct an abstract model $\bar{\mathbf{M}}_1$ utilizing information from the second and third state, that is, let

$$C_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } N_1 = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

and notice that $[0\ 0\ 1] = N_1 C_1$, thus satisfying the condition discussed in Section IV-A. Let $X = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.05 \end{smallmatrix}\right]$ be the solution of the DARE in (10) associated with $\bar{\mathbf{M}}_1$, and observe that $\Sigma_0 - X \succ 0$ and $C_1 X C_1^T \succ 0$ hold in this case, thus implying that

$$\bar{\mathbf{M}}_1 : \begin{cases} \bar{x}_1(t+1) & = \left[\begin{smallmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] \bar{x}_1(t) + \left[\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right] \bar{u}_1(t) + \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] \bar{v}_1(t) \\ \bar{z}_1(t) & = \left[\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}\right] \bar{x}_1(t) \end{cases}$$

where $\bar{x}_1(0) \sim \mathcal{N}(0, \left[\begin{smallmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{smallmatrix}\right])$ and $\bar{v}_1 \sim \mathcal{N}(0, \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0.05 \end{smallmatrix}\right])$, is a time-invariant abstraction for the dynamics in (13).

For $\bar{\mathbf{M}}_1$ to satisfy the specification, take

$$\bar{\mathbf{C}}_1 : \bar{u}_1(t) = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} \bar{x}_1(t)$$

The result will be that $\bar{z}_1(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \bar{v}_1(t+1)$, that is, $\bar{z}_1(t) \sim \mathcal{N}(0, 0.05)$. Using the cumulative distribution function of $\mathcal{N}(0, 0.05)$, we can numerically compute that $\mathbb{P}(\bar{z}_1(t) \notin [-1, 1]) = 7.744\text{e-}6$ for all $t \in [1, 100]$. This implies that $\mathbb{P}(\bar{\boldsymbol{z}}_1 \nvDash \phi) = 7.741\text{e-}4$ and thus that $\mathbb{P}(\bar{\mathbf{C}}_1 \times \bar{\mathbf{M}}_1 \vDash \phi) > 0.95$. Controller $\mathbf{C}_1$ can now be obtained from the controller refinement algorithm described by Algorithm 2. In Fig. 4, the result of applying Algorithm 2 is shown.

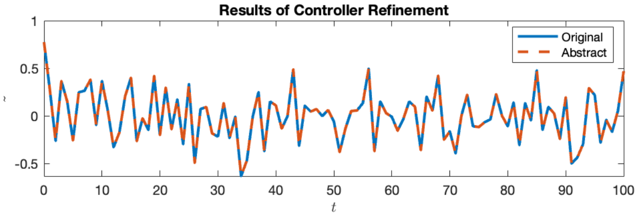

Fig. 4. The performance output of the original- and embedded abstract model over an horizon $t \in [0, 100]$, when applying Algorithm 2.

The above example illustrates that our controller refinement algorithm is sound and the proposed abstraction procedure yields a simplified system, as it leads to model reduction of the dimension of the noise affecting the system. Despite the fact that we considered a simple specification in this example, our correct-by-design results remain unaltered for more complex sc-LTL properties.

It should be noted that the first element of $\bar{x}_1$ only comes into play at the initialization. For $t > 0$, the first element of $\bar{x}_1(t)$ is equal to zero. This leads us to belief that further modification of the current innovation process might yield results for which model order reduction can be applied, to yield equivalent lower order dimensional systems, which can be used to solve the controller synthesis problem more efficient.

Notice that in the example, a trivial case of control design was used. Should any other controller design be used, regardless of the new controller being correct-by-design, the result would be the same, in that the performance outputs, of the original system and the embedded abstract system, would be identical.

Model order reduction.

We now investigate how model reduction can be achieved.

Example 2 (continued from Ex. 1): Let Σ_0 be a stabilizing solution to the DARE (10). Inspired by Lemma 3, consider the system

$$\bar{\mathbf{M}}_2 : \begin{cases} \bar{x}_2(t+1) & = \left[\begin{smallmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] \bar{x}_2(t) + \left[\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right] \bar{u}_2(t) + \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] \bar{v}_2(t) \\ \bar{z}_2(t) & = \left[\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}\right] \bar{x}_2(t) \end{cases}$$

where $\bar{x}_2(0) \sim \mathcal{N}(0, \left[\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.05 \end{smallmatrix}\right])$ and $\bar{v}_2 \sim \mathcal{N}(0, \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0.05 \end{smallmatrix}\right])$.

Due to its structure, system $\bar{\mathbf{M}}_2$ can be realized as

$$\bar{\mathbf{M}}_{2,r} : \begin{cases} \bar{x}_2(t+1) & = \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right] \bar{x}_2(t) + \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right] \bar{u}_2(t) + \bar{v}_2(t) \\ \bar{z}_2(t) & = \left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right] \bar{x}_2(t) \end{cases}$$

where $\bar{x}_2(0) \sim \mathcal{N}(0, \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0.05 \end{smallmatrix}\right])$ and $\bar{v}_2 \sim \mathcal{N}(0, \left[\begin{smallmatrix} 2 & 0 \\ 0 & 0.05 \end{smallmatrix}\right])$.

By taking

$$\bar{\mathbf{C}}_2 : \bar{u}_2(t) = \begin{bmatrix} -1 & 0 \end{bmatrix} \bar{x}_2(t),$$

we can use Algorithm 1 to obtain a correct-by-design controller $\mathbf{C}_2$ using the lower dimensional system $\bar{\mathbf{M}}_{2,r}$.

Important to note is that the above model reduction cannot be quantified by existing simulation relations such as [8], [10], [15]. This makes our method look promising as a new model reduction technique, but for which further research is still necessary.

Notice that in Algorithm 1 a minor modification needs to be made based on the obtained reduced-order model. For instance, in this example, one could simply remove the first element of $\bar{x}(t)$, feeding the remainder to the controller $\bar{\mathbf{C}}_2$.

Lack of Completeness.

The proposed framework is not complete. Due to our choice for the C matrix, when constructing the abstract model, the original problem might become oversimplified to such a degree that no correct-by-design controller can be obtained for the abstract model. However, this failure to design a controller using the generated abstraction does not imply that the temporal property cannot be enforced in the original dynamics.

Example 3 (continued from Ex. 1): Consider again the linear time-invariant stochastic system (13) with specification $\mathbb{P}(\mathbf{C} \times \mathbf{M} \vDash \phi) \geq 0.95$, where $\phi := \Box_{[1,100]} q$. Consider now matrices

$$C_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \text{ and } N_2 = 1,$$

which leads to the following abstract model

$$\bar{\mathbf{M}}_3 : \begin{cases} \bar{x}_3(t+1) & = \left[\begin{smallmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] \bar{x}_3(t) + \left[\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right] \bar{u}_3(t) + \left[\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right] \bar{v}_3(t) \\ \bar{z}_3(t) & = \left[\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}\right] \bar{x}_3(t) \end{cases}$$

where $\bar{x}_3(0) \sim \mathcal{N}(0, \left[\begin{smallmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{smallmatrix}\right])$ and $\bar{v}_3 \sim \mathcal{N}(0, 2.05)$.

For $\bar{\mathbf{M}}_3$, no controller $\bar{\mathbf{C}}_3$ exist such that $\mathbb{P}(\bar{\mathbf{C}}_3 \times \bar{\mathbf{M}}_3 \vDash \phi) \geq 0.95$, since maximizing the probability satisfaction of the property ϕ leads to the controller $\bar{\mathbf{C}}_3 : \bar{u}_3(t) = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} \bar{x}_3(t)$, resulting in $\bar{z}_3(t) \sim \mathcal{N}(0, 2.05)$. Using again the cumulative distribution function, we can numerically compute that $\mathbb{P}(\bar{\mathbf{C}}_3 \times \bar{\mathbf{M}}_3 \vDash \phi) = 1.543\text{e-}29$. This means that there is no controller using $\bar{\mathbf{M}}_3$ that enforces property ϕ for the dynamics (13). But we have already shown how to enforce this property with a more complex abstraction. Hence, Examples 1 and 3 clearly illustrate that while both abstract models simplify the stochastic control synthesis problem, oversimplification may prevent us to find an adequate controller.

VII. Conclusion

In this paper, we established connections between abstractions of discrete-time linear stochastic dynamical systems and the theory of weak Gaussian stochastic realization and Kalman filtering. We proposed a technique that opens the road to model reduction of stochastic processes, whereby a simpler stochastic representation of the original dynamics is obtained by means of Kalman filtering techniques. We also show how to design correct-by-design, feedback controllers for the original dynamics that enforce synthetically co-safe Linear Temporal Logic properties on the closed-loop dynamics, using correct-by-design controllers obtained on the simplified representation. Our controller refinement algorithm is constructive and, under some technical assumptions, our abstraction procedure may lead to time-invariant abstractions of the original dynamics. Finally, some model reduction results were shown which existing abstraction methods, such as those employing simulation relations [8], [10], [15], can not realize.

References

[1] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[2] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[3] C. Belta, B. Yordanov, and E. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 15. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-319-50763-7.pdf

[4] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *17th international IEEE conference on intelligent transportation systems (ITSC)*, 2014, pp. 392–399.

[5] N. Cauchi and A. Abate, "StocHy : Automated verification and synthesis of stochastic processes," in *Tools and Algorithms for the Construction and Analysis of Systems*, 2019, pp. 247–264.

[6] F. Gustafsson, *Statistical sensor fusion*. Studentlitteratur, 2010.

[7] S. Haesaert, N. Cauchi, and A. Abate, "Certified policy synthesis for general markov decision processes: An application in building automation systems," *Performance Evaluation*, vol. 117, pp. 75–103, 2017.

[8] S. Haesaert, S. E. Z. Soudjani, and A. Abate, "Verification of general markov decision processes by approximate similarity relations and policy refinement," *SIAM Journal on Control and Optimization*, vol. 55, no. 4, pp. 2333–2367, 2017.

[9] V. Ionescu and M. Weiss, "Continuous and discrete-time riccati theory: a popov-function approach," *Linear Algebra and its Applications*, vol. 193, pp. 173–209, 1993.

[10] A. A. Julius and G. J. Pappas, "Approximations of stochastic hybrid systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1193–1203, 2009.

[11] V. Krishnamurthy, *Partially observed Markov decision processes*. Cambridge university press, 2016.

[12] A. Lavaei, M. Khaled, S. Soudjani, and M. Zamani, "Amytiss: Parallelized automated controller synthesis for large-scale stochastic systems," in *International Conference on Computer Aided Verification*. Springer, 2020, pp. 461–474.

[13] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, "Automated verification and synthesis of stochastic hybrid systems: A survey," *Automatica*, vol. 146, p. 110617, 2022.

[14] P. Nilsson, S. Haesaert, R. Thakker, K. Otsu, C.-I. Vasile, A.-A. Agha-Mohammadi, R. M. Murray, and A. D. Ames, "Toward specification-guided active mars exploration for cooperative robot teams," *Robotics: Science and Systems*, vol. XIV, pp. 1–9, 2018.

[15] G. Pola, C. Manes, A. J. van der Schaft, and M. D. Di Benedetto, "Bisimulation equivalence of discrete-time stochastic linear control systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1897–1912, 2017.

[16] S. E. Z. Soudjani, C. Gevaerts, and A. Abate, "FAUST^2: Formal abstractions of uncountable-state stochastic processes," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, 2015, pp. 272–286.

[17] M. Svoreňová, M. Chmelík, K. Leahy, H. F. Eniser, K. Chatterjee, I. Černá, and C. Belta, "Temporal logic motion planning using POMDPs with parity objectives," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 2015, pp. 233–238.

[18] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[19] B. van Huijgevoort, O. Schön, S. Soudjani, and S. Haesaert, "SySCoRe: Synthesis via stochastic coupling relations," *arXiv preprint arXiv:2302.12294*, 2023.

[20] J. H. van Schuppen, *Control and System Theory of Discrete-Time Stochastic Systems*. Springer, 2021.

Appendix

Additional information on the derivations for (6) *and* (7).

We have that $\hat{x}(t) = x_K(t|t-1)$, $\bar{x}(t) = x_K(t|t)$, $\hat{P}(t) = P(t|t-1)$ and $\bar{P}(t) = P(t|t)$. The innovation processes are obtained by simple substitution of the equations (4) and the innovation (5). The initial state distribution $\bar{x}(0)$ is obtained from (4a), $y(0) = Cx(0)$, $x(0) \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $K(0) = \Sigma_0 C^T[C\Sigma_0 C^T]^{-1}$. $\bar{P}(0)$ is obtained from (4c).

Proof: [Proof of Theorem 1] The proof follows directly from the following lemmas and proposition.

Lemma 7: $\mathbf{M}$ and $\mathbf{M}_{\text{Obs}}$ have the same performance output distributions when given the same input.

This follows directly from $NC = H$.

Proposition 8: Assume that $C\Sigma_0 C^T \succ 0$ and $CQ_w C^T \succ 0$. $\mathbf{M}_{\text{Obs}}$ and $\hat{\mathbf{M}}$ have the same performance output distributions $\mathbf{z}$ and $\hat{\mathbf{z}}$ if $u(t) = \hat{u}(t), \ \forall t \in \mathbb{N}_0$.

The above proposition is mostly equivalent to Prop. 8.4.3 in [20], the main differences being the additional input and output in $\mathbf{M}_{\text{Obs}}$ and $\hat{\mathbf{M}}$, and the absence of output disturbance in $\mathbf{M}_{\text{Obs}}$. The former two, however, do not influence the result, as can be observed from comparing [20, Theorem 8.3.2 & Theorem 14.4.2] with the former being the main contributor to the proof of Prop. 8.4.3. On the other hand, the exclusion of output disturbance in $\mathbf{M}_{\text{Obs}}$ is resolved by assuming that $C\Sigma_0 C^T \succ 0$ and $CQ_w C^T \succ 0$.

Lemma 9: $\hat{\mathbf{M}}$ and $\bar{\mathbf{M}}$ have the same performance output distributions $\hat{\mathbf{z}}$ and $\bar{\mathbf{z}}$ if $\hat{u}(t) = \bar{u}(t), \ \forall t \in \mathbb{N}_0$.

Proof: Since the innovation processes can be obtained from one-another by direct substitution of the Kalman filter equations (4) and the innovation (5), without changing their performance outputs, both have the same performance outputs $\hat{\mathbf{z}}$ and $\bar{\mathbf{z}}$ if $\hat{u}(t) = \bar{u}(t), \ \forall t \in \mathbb{N}_0$. □

Accordingly, the proof of the theorem follows. □