
Tracking of Dynamic Physical Systems with Dynamic Bayesian Net Models Factored using Structural Observability

Abstract

Monitoring, diagnosis, and control of complex physical systems require tracking their dynamic behavior during system operation. Effective tracking requires an accurate system behavior model, and sufficient measurements to estimate system state. However, models suffer from lack of accuracy, and measurements are noisy, making the tracking task difficult. Dynamic Bayes Nets (DBNs) provide a general framework for modeling and reasoning about complex systems under uncertainty. However, reasoning with DBNs can be computationally expensive. This paper presents an approach to improve the tracking efficiency by partitioning the system DBN into smaller, conditionally independent DBN factors. We present a methodology of deriving the DBN factors by analyzing the structural observability of dynamic systems and demonstrate the effectiveness of our factoring approach for accurate, yet more efficient, tracking of dynamic systems.

1 Introduction

Monitoring, diagnosis, and control of complex systems require tracking their dynamic behavior during system operation. Effective tracking requires an accurate model of system behavior, and sufficient measurements to estimate system state. Moreover, to be useful in real-world scenarios, the tracking approaches must account for process and observation noise. Hence, many tracking approaches are stochastic (Murphy, 2002).

In general, stochastic tracking approaches use a discrete-time invariant nonlinear dynamic system model: $\mathbf{X}_{t+1} = f(\mathbf{X}_t, \mathbf{U}_t) + \mathbf{V}_t$, and $\mathbf{Y}_{t+1} = h(\mathbf{X}_{t+1}) + \mathbf{W}_{t+1}$, where, \mathbf{X} , \mathbf{U} , and \mathbf{Y} denote vectors of (hidden) state, input, and measured random variables in

the dynamic system, respectively, and \mathbf{V} and \mathbf{W} denote vectors of process and observation noises, respectively. Subscript t represents a variable at time t . Variables \mathbf{X} , \mathbf{Y} , and \mathbf{U} are also considered stochastic random variables, and a Gaussian distribution is assumed around each random variable, and the functions $f(\cdot)$ and $h(\cdot)$ are assumed to be deterministic. Given this formulation, the *tracking* problem involves determining $P(\mathbf{X}_{t+1} | \mathbf{Y}_{0:t+1})$.

For linear time-invariant (LTI) systems, Kalman filters (KFs) (Murphy, 2002) can be used to solve the tracking problem recursively, under the assumption that the process and observation noises are zero mean multivariate Gaussian noises with covariance \mathbf{Q} and \mathbf{R} , respectively. A LTI system is modeled as: $\mathbf{X}_{t+1} = \mathbf{A}\mathbf{X}_t + \mathbf{B}\mathbf{U}_t + \mathcal{V}\mathbf{V}_t$, and $\mathbf{Y}_{t+1} = \mathbf{C}\mathbf{X}_{t+1} + \mathcal{W}\mathbf{W}_{t+1}$, where, \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathcal{V} , and \mathcal{W} are matrices with appropriate dimensions. Given this formulation, the behavior tracking using Kalman Filter is set up in two phases. The *prediction* phase involves: $\hat{\mathbf{X}}_{t+1}^- = \mathbf{A}\hat{\mathbf{X}}_t^- + \mathbf{B}\mathbf{U}_t$, and $\mathbf{P}_{t+1}^- = \mathbf{A}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}$, where, \mathbf{P}^- and \mathbf{P} denote a *priori* and a *posteriori* estimate error covariance, respectively, and $\hat{\mathbf{X}}^-$ and $\hat{\mathbf{X}}$ denote a *priori* and a *posteriori* state estimates, respectively. The *update* phase of KF involves: $\mathbf{K}_{t+1} = \mathbf{P}_{t+1}^- \mathbf{C}^T (\mathbf{C}\mathbf{P}_{t+1}^- \mathbf{C}^T + \mathbf{R})^{-1}$, $\hat{\mathbf{X}}_{t+1} = \hat{\mathbf{X}}_{t+1}^- + \mathbf{K}_{t+1}(\mathbf{Y}_{t+1} - \mathbf{C}\hat{\mathbf{X}}_{t+1}^-)$, and $\mathbf{P}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{C})\mathbf{P}_{t+1}^-$, where \mathbf{K} is the Kalman gain.

In order for the KF to give accurate state estimates, the system must be *observable*.

Definition 1. (Samantaray and Bouamama, 2008) (Observability). A system is *observable* if its initial state variables, \mathbf{X}_{t_0} , at time t_0 , can be derived from the knowledge of inputs, $\mathbf{U}_{t_0:t_f}$, and outputs, $\mathbf{Y}_{t_0:t_f}$, in the time interval $[t_0, t_f]$, where t_f is the current time.

A KF built around a system with unobservable states will simply not work, because, by definition, an unobservable state is one about which no information can be obtained using the measurements. An n^{th} -order LTI system is observable if its observability matrix,

$\mathcal{O} = [\mathcal{C}^t, (\mathcal{C}\mathcal{A})^t, \dots, (\mathcal{C}\mathcal{A}^{n-1})^t]^t$ is of full rank, i.e., $\text{rank}(\mathcal{O}) = n$.

KFs can be extended to Extended Kalman Filters (EKFs) for nonlinear system, under the assumption of zero-mean multivariate Gaussian distributions for process and measurement error (Murphy, 2002). Dynamic Bayesian Networks (DBNs) provide the most general method for modeling the dynamics of complex systems in the presence of noise and sensor inaccuracies (Murphy, 2002). DBN-based state-estimation methods apply to nonlinear systems and arbitrary probability distributions, and hence, generalize KFs and EKFs. A DBN is a 2-slice Bayesian network, that compactly models a dynamic system. The nodes of a DBN represent random variables, and directed links capture the causal relations between these variables at a time point, and across consecutive time steps. However, inference algorithms using DBNs is exponential in the number of state variables, and for nonlinear systems and non-Gaussian noise models, analytic, closed form, exact estimation methods may not exist. Approximate estimation algorithms, e.g., particle filters (PF) (Koller and Lerner, 2001), are therefore used for tracking dynamic behavior and state estimation. However, these approaches also require large computational resources.

This paper proposes an approach to improve the computational efficiency of DBN-tracking schemes by partitioning the DBN into factors that represent independent subsystems conditioned on a set of measurements, and applying tracking algorithms to each factor independently. Specifically, we generate DBN factors by expressing some of the state variables as algebraic functions of measurements converted to system inputs. As a result, the across-time links directed to wards these state variables are replaced by intra-time links from the measurements. This makes some of the state variables in the generated factors conditionally independent from the state variables in other factors, given the selected measurements. Hence, the state estimation in one factor becomes independent of the other factors, and the state estimation for the individual factors can be carried out separately, thus increasing the overall computational efficiency.

Recall that estimation of state variables from the system measurements works only if the physical system is observable. The traditional control-system-theoretic schemes for analyzing observability apply to linear systems and depend on the numerical values of the system parameters. The approach presented in this paper employs the bond graph (BG) modeling paradigm (Karnopp et al., 2000) to establish *structural observability* (Sueur and Dauphin-Tanguy, 1991, 1989) for each DBN factor. Structural observability does not

depend on the numerical values of the system parameters, and applies to nonlinear systems where the nonlinearities are in the system components, and not in the system structure.

2 Related Work

Our distributed inference approach, applicable to continuous-time, dynamic, physical systems, improve upon other distributed inference schemes existing in literature, such as, Boyen-Koller (BK) algorithm (Boyen and Koller, 1998) and Distributed decentralized extended Kalman filters (DDEKF) (Mutambara, 1998). The BK algorithm creates the individual factors by eliminating causal links between weakly interacting subsystems. The belief state derived from the individual factors is an approximation of the true belief state, but the error in approximation is bounded. However, the bounds may not be accurate for inference of physical systems. Heuristic techniques for automatically decomposing a DBN into factors are presented in (Frogner and Pfeffer, 2008). This approach results in lower estimation errors, but the computed factored belief state is still an approximation. The Factored Particle filtering (FPF) scheme (Ng and Peshkin, 2002) further reduces estimation errors by applying the PF scheme to the BK factored inference approach. Our factoring scheme is not arbitrary, but a result of thorough analysis of structural observability, and the factors exactly preserve the overall system dynamics in factored form, and our estimation approach uses the PF scheme on each factor. Hence, we produce accurate state estimates efficiently.

Moreover, in BK algorithm, the results from different factors are integrated together to get globally correct inference, thereby still having the issue of single point of failure. DDEKFs represent an approach for subdividing the estimation problem into smaller subproblems. However, in DDEKFs, each local component requires both measurements and state variable estimate from other components to correctly estimate its states. As a result, inaccuracies in one component can affect the estimation in other components. Our estimation approach for each individual factor is more robust to failures in other factors (as long as the required measurements are available) because the random variables in a factor are conditionally independent of those in all other factors, given a subset of the measurements.

3 Modeling Dynamic Systems using DBNs

A two-step systematic procedure for generating DBNs for physical systems has been developed: (i) gener-

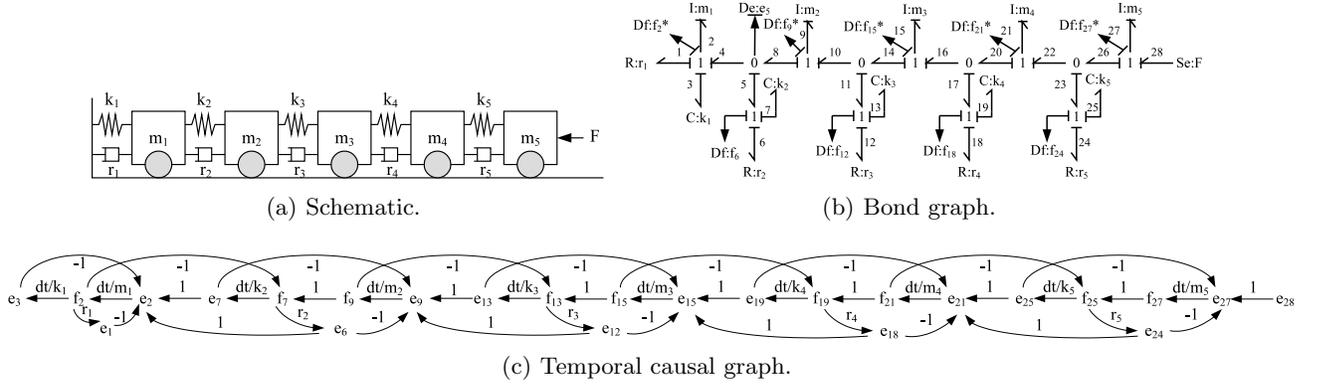


Figure 1: Spring-mass-damper system models.

ate a temporal causal graph (Mosterman and Biswas, 1999) from the bond graph (BG) model of the system, and (ii) generate a DBN from the TCG (Lerner et al., 2000).

The *bond graph* (BG) modeling paradigm provides a framework for domain-independent, energy-based, topological modeling of physical processes. The nodes of a BG include energy storage (capacitors, C , and inertias, I), dissipation (resistors, R), transformation (gyrators, GY , and transformers, TF), source (effort sources, Se , and flow sources, Sf), and detection (effort detectors, De , and flow detectors, Df) elements. Nonlinear systems are modeled by modulated versions of BG elements whose values are functions of other system variables (e.g., MSe , denotes a “modulated Se ”).

Bonds, drawn as half arrows, with associated effort, e , and flow, f , variables, represent the power interaction pathways between the bond graph elements, such that $e \times f$ defines the power transferred through the bond. 0- and 1-junctions represent idealized connections for lossless energy transfer between two or more BG elements.

Fig. 1(b) shows the BG of a simple tenth-order spring-mass-damper (SMD) system (shown in Fig. 1(a)). In the mechanical domain, I elements are masses, C elements are springs, R elements are dampers, flows represent velocities (e.g., f_2 denotes the velocity of mass m_1 , and f_6 denotes the relative velocity between masses m_2 and m_1), and efforts represent forces, (e.g, e_3 denotes the force on spring k_1). The velocities f_6^* , f_9^* , f_{12}^* , f_{15}^* , f_{18}^* , f_{21}^* , f_{24}^* , f_{27}^* and the force e_5 are the available sensors (measurements) in this SMD model. The force, e_{28} , impressed upon mass m_5 is a system input. In this system, the BG parameters are assumed to be constant.

The *temporal causal graph* (TCG) of a system, systematically derived from its BG (Mosterman and Biswas, 1999), captures the causal and temporal relations be-

tween system variables through directed *edges* and their *labels*. *Causality* establishes the cause and effect relationships between the e and f variables of the bonds determined by constraints imposed by the incident BG elements. Of special interest are the energy storage elements, which can either impose *integral* (preferred) or *derivative* causality. The sequential causal assignment procedure (SCAP) systematically assigns the causality in a BG (Karnopp et al., 2000). The nodes in a TCG correspond to the power variables of the system BG model. Fig. 1(c) shows the TCG for the SMD system. The direction of a TCG edge and its label are based on causality. For example, for a C element in integral causality, $e = (1/C) \int f dt$, and hence the TCG edge directed from the flow to the effort has a label dt/C , with dt denoting a temporal relationship between f and e . For a C element in derivative causality, the TCG edge is directed from the effort to the flow, since $f = Cde/dt$, and has a label C/dt . The system DBN can be constructed from its TCG in integral causality using the method outlined in (Lerner et al., 2000).

(Murphy, 2002) defines a DBN as $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, where \mathbf{X} , \mathbf{U} , and \mathbf{Y} are sets of stochastic random variables that denote (hidden) state variables, system input variables, and measured variables in the dynamic system, respectively. Graphically, a DBN is a two-slice Bayesian network, representing a snapshot of system behavior in two consecutive time slices, t and $t + 1$. Each DBN time-slice represents the Markov process observation model, $P(\mathbf{Y}_t | \mathbf{X}_t, \mathbf{U}_t)$ derived from causal links $X_t \rightarrow Y_t$ and $U_t \rightarrow Y_t$, where $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, $U \in \mathbf{U}$, and subscript t represents a variable at time t . Similarly, across-time causal links $X_t \rightarrow X_{t+1}$, $X_t \rightarrow X'_{t+1}$, and $U_t \rightarrow X_{t+1}$, where $X' \in \mathbf{X}$, represent the Markov state-transition model, $P(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{U}_t)$.

After we identify the TCG nodes, \mathbf{N} , which include all state variables, measured variables, and system inputs; for each $N \in \mathbf{N}$, we instantiate nodes N_t and N_{t+1} in

the consecutive time slices of the DBN. Then, for every pair of variables, $N, N' \in \mathbf{N}$ that are algebraically related, causal links $N_t \rightarrow N'_t$ and $N_{t+1} \rightarrow N'_{t+1}$ are constructed in each DBN time slice. For every pair of variables, $N, N' \in \mathbf{N}$ having an integrating relation (i.e., a delay), the across-time $N_t \rightarrow N'_{t+1}$ link is added to the DBN. Fig. 2(a) shows the DBN for the spring-mass-damper (SMD) system, where thick-lined circles denote state variables, thin-lined circles denote observed variables, and squares denote input variables. If a state variable N is directly measured, two new nodes N_t^* and N_{t+1}^* are created in the DBN, and links $N_t \rightarrow N_t^*$ and $N_{t+1} \rightarrow N_{t+1}^*$ are established. For example, since the state variable f_2 of SMD is also measured, the SMD DBN has nodes f_{2t}^* and f_{2t+1}^* .

4 State Estimation Using Particle Filters

The general iterative solution of the DBN state estimation problem is $P(\mathbf{X}_{t+1} | \mathbf{Y}_{0:t+1}) = \alpha P(\mathbf{Y}_{t+1} | \mathbf{X}_{t+1}, \mathbf{U}_t) \times \sum_{\mathbf{X}_t} P(\mathbf{X}_{t+1} | \mathbf{X}_t, \mathbf{U}_t) P(\mathbf{X}_t | \mathbf{Y}_{0:t})$, where $\mathbf{Y}_{0:t}$ denotes measurement readings from time 0 to t , and α is the normalizing factor (Murphy, 2002). In this work, we choose PF as our iterative algorithm for DBN state estimation (Koller and Lerner, 2001). PF is a sequential Monte Carlo method that approximates the belief state of a system using a weighted set of samples, or *particles* (Arulampalam et al., 2002). The value of each particle describes a possible system state, and its weight denotes the likelihood of the observed measurements given this particle's value. As more observations are obtained, each particle is moved stochastically to a new state, and the weight of each particle is readjusted to reflect the likelihood of that observation given the particle's new state.

5 Structural Observability

To ensure accurate tracking of system behavior for diagnosis, the system must be *observable*, i.e., all its state variables can be correctly determined given the available measurements (Samantaray and Bouamama, 2008). We describe a more general property of structural observability, and show how this property holds for nonlinear systems. Recall that an n^{th} -order LTI system is observable if its observability matrix, $\mathcal{O} = [\mathcal{C}^t, (\mathcal{C}\mathcal{A})^t, \dots, (\mathcal{C}\mathcal{A}^{n-1})^t]^t$ is of full rank, i.e., $\text{rank}(\mathcal{O}) = n$. Therefore, system observability is a function of the numeric values of the system parameters. *Structural observability* alternatively defines observability as a function of the system structure (Sueur and Dauphin-Tanguy, 1991, 1989). This notion of *structural observability* holds for a class of

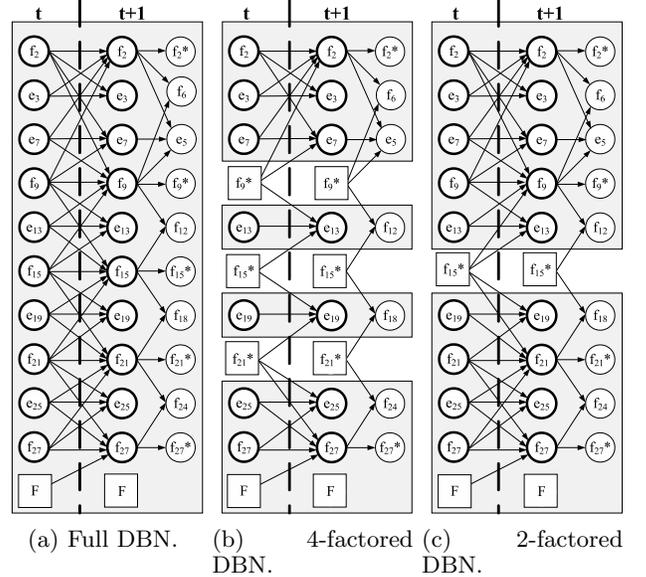


Figure 2: Factorings of the Spring-mass-damper DBN.

structurally equivalent systems. If a system is structurally observable, but its \mathcal{O} matrix is not of full rank, i.e., $\text{rank}(\mathcal{O}) < n$, the full rank can be restored by perturbing the values of elements of its \mathcal{A} and \mathcal{C} matrices (Sueur and Dauphin-Tanguy, 1991). The structural observability properties of a system can be determined by analyzing its BG (Sueur and Dauphin-Tanguy, 1991). The notion of *structural rank* (*struct-rank*) is central to this analysis.

Definition 2. (Sueur and Dauphin-Tanguy, 1991) (Structural Rank). *Structural rank* of a matrix is defined as the maximal rank of this matrix as a function of its free parameters, taking into account the relations between parameters.

For example, $\text{struct-rank} \left(\begin{bmatrix} -R/L_1 & R/L_2 \\ R/L_1 & -R/L_2 \end{bmatrix} \right) = 1$, since the second row of the matrix is linearly dependent on the first row.

Given the BG model of a system with matrices \mathcal{A} , \mathcal{B} , \mathcal{C} , and \mathcal{D} , the system is structurally observable iff (Sueur and Dauphin-Tanguy, 1991): (1) every dynamical element of the BG in integral causality is causally connected to a measurement sensor, and (2) $\text{struct-rank}([\mathcal{A}^t \ \mathcal{C}^t]^t) = n$, where n is the number of state variables in the system.

Intuitively, condition 1 is satisfied if for each independent decoupled subsystem, at least one dynamical element in integral causality is causally connected to a measurement. Condition 2 is satisfied if the causality of *every* I and C element initially in integral causality can be inverted to produce a valid derivative causality assignment for the BG model. In some situations, De

and Df elements may have to be changed into their dual form to assign consistent derivative causality to the BG. This procedure for manipulating the BG to directly determine the structural rank of matrix $[\mathcal{A}^t \mathcal{C}^t]^t$ has been presented in (Sueur and Dauphin-Tanguy, 1991).

The proposed method for analyzing structural observability for linear systems can be extended for nonlinear systems when the nonlinearities can be expressed by making the I , C , and R values as functions of other variables, since the notion of junction structure remains unchanged from that of linear systems. However, this equivalence does not hold when the nonlinearities are linked to the system structure (Sueur and Dauphin-Tanguy, 1991).

6 Factoring DBNs For Efficient State Estimation

6.1 Problem Statement

Given a DBN $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, our goal is to factor D into the *maximal* number of conditionally independent *DBN Factors* (DBN-Fs), $D_i = (\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i)$, $i \in [1, m]$, such that each DBN-F is *observable*. Observability and conditional independence of each DBN-F is a necessary condition for ensuring efficient and accurate state estimates when the estimation algorithm is applied to each DBN-F separately.

Definition 3 (DBN Factor). A *DBN Factor* (DBN-F), $D_i = (\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i)$, $i \in [1, m]$, of DBN $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$ is a smaller DBN such that (i) $\bigcup \mathbf{X}_i \subset \mathbf{X}$, (ii) $\bigcup \mathbf{Y}_i \subset \mathbf{Y}$, (iii) $\bigcup \mathbf{U}_i = \mathbf{U} \cup (\mathbf{Y} - \bigcup \mathbf{Y}_i)$, and (iv) each D_i is *conditionally independent* from all other DBN-Fs given the inputs, \mathbf{U}_i .

Definition 4 (Conditionally Independent DBN-F). Any DBN-F, $D_j = (\mathbf{X}_j, \mathbf{U}_j, \mathbf{Y}_j)$, of a global DBN, $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, is *conditionally independent* from all its other DBN-Fs $D_k = (\mathbf{X}_k, \mathbf{U}_k, \mathbf{Y}_k)$, s.t. $k \neq j$, $k \in [1, m]$ given \mathbf{U}_j if (i) $P(\mathbf{X}_{j,t+1} | \mathbf{X}_{t-n:t}, \mathbf{U}_{t-n:t}) = P(\mathbf{X}_{j,t+1} | \mathbf{X}_{j,t-n:t}, \mathbf{U}_{j,t-n:t})$, and (ii) $P(\mathbf{Y}_{j,t} | \mathbf{X}_t, \mathbf{U}_t) = P(\mathbf{Y}_{j,t} | \mathbf{X}_{j,t}, \mathbf{U}_{j,t})$.

Definition 5 (Observable DBN-F). A DBN-F, $D_j = (\mathbf{X}_j, \mathbf{U}_j, \mathbf{Y}_j)$ is *observable* if the underlying subsystem it represents is *structurally observable*.

Example: Fig. 2(b) shows four DBN-Fs, $D_1 = (\{f_2, e_3, e_7\}, \{f_9^*\}, \{f_2^*, f_6, e_5\})$, $D_2 = (\{e_{13}\}, \{f_9^*, f_{15}^*\}, \{f_{12}\})$, $D_3 = (\{e_{19}\}, \{f_{15}^*, f_{21}^*\}, \{f_{18}\})$, and $D_4 = (\{e_{25}, f_{27}\}, \{f_{21}^*, F\}, \{f_{24}, f_{27}^*\})$. As shown in Defn. 3, $\bigcup_{i \in [1,4]} \mathbf{X}_i \subset \mathbf{X}$, $\bigcup_{i \in [1,4]} \mathbf{Y}_i \subset \mathbf{Y}$, $\bigcup_{i \in [1,4]} \mathbf{U}_i = \mathbf{U} \cup (\mathbf{Y} - \bigcup \mathbf{Y}_i)$. Also, each DBN-F shown in Fig. 2(b) is conditionally independent of all other DBN-Fs. For example, in the global DBN shown in Fig. 2(a), the

value of e_{13} at time step $t + 1$ depends on f_9 , e_{13} , and f_{15} at time step t , and f_2 and e_7 , among others, at time step $t - 1$, and so on. However, DBN-F, D_2 , shown in Fig. 2(b), is conditionally independent of all other DBN-Fs given its inputs f_9^* and f_{15}^* because the values of its state variable, e_{13} , and measurement variable, f_{12} , at time t , do not depend on any variable external to D_2 .

6.2 Overview of Factoring Approach

Our procedure for *factoring* a DBN involves replacing one or more of its state variables by algebraic functions of at most r measured variables, \mathbf{Y}^r , where r is a user-specified parameter. Once we express a state variable in terms of \mathbf{Y}^r , i.e., $X = g^{-1}(\mathbf{Y}^r)$, considering \mathbf{Y}^r to be inputs, we delete every $X_t \rightarrow X_{t+1}$, $U_t \rightarrow X_{t+1}$, $X_t \rightarrow Y_t$ link, and replace X with $g^{-1}(\mathbf{Y}^r)$. Then, we restore an intra-time slice link $g^{-1}(\mathbf{Y}^r) \rightarrow Y_t$ for every $X_t \rightarrow Y_t$, such that $Y_t \notin \mathbf{Y}^r$. The across-time links into X_t are not restored, since $g^{-1}(\mathbf{Y}^r)$ can be computed independently at each time step. The replacing of sufficient number of state variables in terms of measurements, and the subsequent removal of across-time links involving these state variables produces conditionally independent DBN-Fs.

The goal of our factoring scheme is to generate the maximal number of DBN-Fs that are each observable. A DBN can be factored into maximal number of *observable* DBN-Fs by (i) generating maximal number of (possibly unobservable) conditionally independent factors by replacing every state variable which can be determined as an algebraic function of at most r measurements, and (ii) merging unobservable DBN-Fs from this maximal factoring into other factors till all of the generated factors are observable.

Example: For the DBN shown in Fig. 2(b), assuming $r = 1$, measurements, f_2^* , f_9^* , f_{15}^* , f_{21}^* , and, f_{27}^* , each depend on the single state variable, f_2 , f_9 , f_{15} , f_{21} , and, f_{27} , respectively. In this system, f_{15} is directly measured, so $f_{15}^* = g(f_{15})$ trivially exists, and so does the function $h = g^{-1}$. (More generally, a set of measured variables may be needed to establish the value of a state variable, and h will be a function derived from f and g .) Hence, as shown in Fig. 2(b), if we replace f_{15} with the measurement, f_{15}^* , we no longer need variables f_9 , e_{13} , f_{15} , e_{19} , and f_{21} to compute f_{15} . Thus the across-time links to f_{15} can be removed. So, given the measurement f_{15}^* , the DBN-Fs $D_2 = (\{e_{13}\}, \{f_9^*, f_{15}^*\}, \{f_{12}\})$ and $D_3 = (\{e_{19}\}, \{f_{15}^*, f_{21}^*\}, \{f_{18}\})$ are conditionally independent. Repeating the above procedure and replacing f_9 and f_{21} yields the maximally factored SMD DBN, shown in Fig. 2(b), which contains 4 DBN-Fs. The two middle DBN-Fs in Fig. 2(b) are not observable, since

Algorithm 1 Generating factors of a DBN.

Input: System DBN, D
 Generate maximal $Factoring_1 = \{D_1, D_2, \dots, D_n\}$
 $SetOfFactorings = \{Factoring_1\}$
while true do
 $SetOfObsF = \emptyset$; $SetOfUnobsF = \emptyset$;
 for each $Factoring_i \in SetOfFactorings$ **do**
 if every DBN-F in $Factoring_i$ is observable **then**
 $SetOfObsF = SetOfObsF \cup Factoring_i$
 else
 $SetOfUnobsF = SetOfUnobsF \cup Factoring_i$
 if $SetOfObsF \neq \emptyset$ **then**
 $BestFactoring = Factoring_j \in SetOfObsF$ having
 the most number of balanced DBN-Fs
 exit
 else
 $NextBestFactoring = Factoring_j \in$
 $SetOfUnobsF$ having the most number of
 unobservable DBN-Fs
 $SetOfFactorings =$ all possible pairwise mergings of
 the DBN-Fs of $NextBestFactoring$

the single state variable in either of the two DBN-Fs does not affect the observed variable. However, the factoring generated by merging each unobservable DBN-F to its observable neighbor (see Fig. 2(c)) results in a factoring where all DBN-Fs are observable.

6.3 The Factoring Algorithm

Our algorithm for generating maximal number of observable DBN-Fs from a given DBN is as follows: (i) partition the DBN into maximal DBN-Fs, (ii) map each generated DBN-F to a BG fragment (BG-F) and analyze the structure of this BG-F to determine if the DBN-F is observable, and (iii) merge every unobservable DBN-F with other DBN-Fs so as the resultant DBN-Fs may be observable, till all DBN-Fs are observable. These steps (shown in Algorithm 1) are presented in detail below. We assume that the system to be factored is observable, as otherwise, no factoring with only observable factors exist. Also, we assume that we have sufficient sensors to allow factoring.

6.3.1 Step 1 - Generating Maximal Factoring

Given the user-specified parameter, r , we analyze the system DBN to identify all state variables that are algebraic functions of single measurements, or pairs of measurements, or triples, and so on, up to r measurements. Then we express these state variables in terms of a subset of measurements, remove the state variables, and all across-time links directed into them. The maximally factored DBN for the SMD system is shown in Fig. 2(b). However, a state variable is not expressed in terms of measurements if the removal of this state variable does not generate any new factors. For example, in Fig. 2(b), f_2 is not replaced with measure-

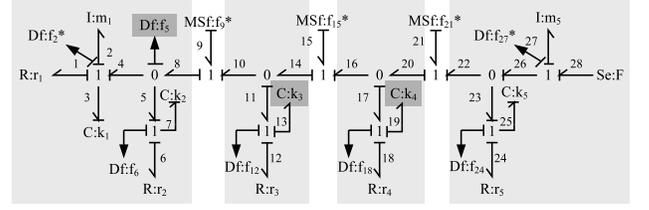


Figure 3: Four-Factored SMD bond graph with imposed derivative causality.

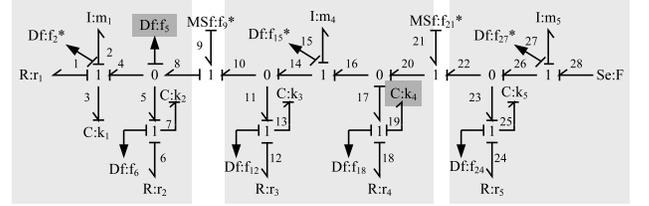


Figure 4: Three-Factored SMD bond graph with imposed derivative causality.

ment f_2^* since this does not generate any new factors.

6.3.2 Step 2 - Testing Observability of DBN-Fs

Given a DBN-F D_i , we can test whether or not it is observable by first mapping D_i to a BG-F, and analyzing this BG-F, B_i for structural observability. Before mapping a D_i to a B_i , we identify the state variables in the global DBN that were removed to generate D_i , and the measurement variables these state variables were replaced with. Given this information, the first step of mapping a D_i to a B_i is to replace the I or C element (in the global BG) corresponding to every state variable that was removed from the global DBN to generate D_i by a MSf or MSe element, respectively, whose value is computed in terms of at most r measurements. Then, we define B_i to be that fragment of the system BG that lies between these newly introduced MSf or MSe elements, as the BG is factored into independent subsystems by these source elements.

Proposition 1. A BG may be factored into *independen-*

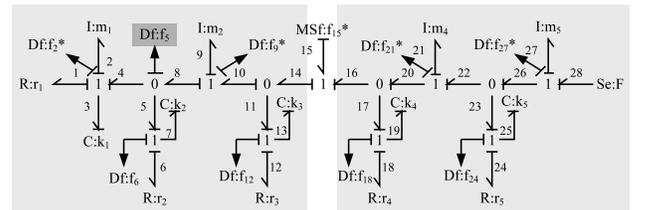


Figure 5: Two-Factored SMD bond graph with imposed derivative causality.

dent BG factors B_1, B_2, \dots, B_n by replacing an I or C element with a MSf or MSe element, respectively.

Proof: A capacitor C_1 's constituent equation is $e_{C_1} = \frac{1}{C_1} \int f_{C_1} dt$. In the state-space formulation, f_{C_1} can be expressed in terms of other state variables. Hence, any measurement or state variable that depends on e_{C_1} would, in turn, be dependent on f_{C_1} , and possibly every other state variable. Now, if f_{C_1} can be measured, and we replace C_1 with modulated $MSe_{C_1} = g^{-1}(f_{C_1})$, the dependence between e_{C_1} and all other state variables is broken, and the BG is factored into *independent* BG-Fs. The proof similarly follows for an I element replaced with a modulated MSf . ■

Example: The maximally factored SMD DBN has four DBN-Fs (Fig. 2(b)), which correspond to the BG-Fs shown in Fig. 3. The two outer BG-Fs are structurally observable, as all their energy storage elements can be assigned preferred derivative causality (albeit by dualizing an effort sensor into a flow sensor, indicated by the shaded background, in the first BG-F), and every state variable affects at least one sensor. The two BG-Fs in the middle, however, are not observable, since, in each of these two BG-Fs, the single state variable does not causally affect the flow sensor. Hence, the maximal DBN factoring shown in Fig. 2(b) cannot be used for accurate state estimation, and some of the factors need to be merged to generate observable DBN-Fs.

6.3.3 Step 3 - Merging Unobservable Factors

Unobservable DBN-Fs can be *merged* with other DBN-Fs to generate an observable DBN-F. m DBN-Fs, D_1, D_2, \dots, D_m , can be merged by restoring those state variables and across-time links in the system DBN that were replaced to generate D_1, D_2, \dots, D_m . The measurements that were used to compute these state variables are also reintroduced.

Merging an unobservable DBN-F, D_1 , with another DBN-F, D_2 , results in DBN-F, $D_{1,2}$, which maps to the BG-F, $B_{1,2}$. The merging of D_1 and D_2 results in the replacement of at least one source element in B_1 and B_2 with a I or C element, and reintroduction of at least one sensor element in the resultant $B_{1,2}$. Since, the reintroduced measurement sensors are directly connected to the reinstated I or C elements in $B_{1,2}$, condition 1 of structural observability is satisfied for these reintroduced energy storage elements. Moreover, the new sensor can be causally linked to other I or C elements that are not linked to any sensor element, further aiding the satisfaction of condition 1 for $B_{1,2}$. Also, the greater are the number of sensors in $B_{1,2}$, the greater is the flexibility for dualizing these sensors to satisfy condition 2.

Algorithm 1 shows how the merging procedure is invoked if a DBN-F in the maximally factored DBN is not observable. In each iteration of this algorithm, we create new factorings through all possible pairwise mergings of unobservable DBN-Fs, to create at least one new factoring with all its DBN-Fs observable. When multiple factorings are generated, we use a heuristic to choose that factoring which has the most number of *balanced* DBN-Fs with respect to state variables. If the merging step does not generate any factorings with all its DBN-Fs observable, we select the maximal factoring with the largest number of factors and highest number of unobservable DBN-Fs, and generate the next set of factorings by pairwise merging of unobservable DBN-Fs. This procedure is repeated till we obtain at least one factoring where all the DBN-Fs are observable. Since the system was initially observable, continued merging will eventually result in a factoring in which all DBN-Fs are observable, at worst producing the original DBN model. Therefore, our factoring algorithm terminates.

Example: The unobservable DBN-Fs, shown in Fig. 2(b), can be merged in two possible ways to form two different factorings. The factoring shown in Fig. 4 corresponds to a DBN-F generated by merging the two central DBN-Fs, and is not unobservable (since capacitor k_4 does not provide a consistent causal assignment when it is assigned derivative causality). However, the two BG-Fs shown in Fig. 5, and corresponding to the DBN-Fs shown in Fig. 2(c), are observable, and hence, we select this as our desired factoring.

6.4 Tracking using Factored DBNs

Given m observable DBN-Fs, D_1, D_2, \dots, D_m , we can implement an inference algorithm on each DBN-F as an independent process. In this work, we implemented m PFs, one for each DBN-F. Each PF takes as inputs, \mathbf{U}_i , and estimates \mathbf{X}_i based on \mathbf{Y}_i . Only measurements $\bigcup_i \mathbf{U}_i$ are communicated between PFs. The PF for the DBN-F D_i uses $a \frac{|\mathbf{X}_i|}{|\mathbf{X}|}$ particles, where a is a user-specified parameter. For m DBN-Fs, $\sum_i |\mathbf{X}_i| < |\mathbf{X}|$, where \mathbf{X} is the total number of states in the complete system. Therefore, the complexity of tracking using each DBN-F is less than that of tracking using the global DBN. Also, since the inference algorithms on the different factors are executed simultaneously, the total complexity of inference reduces to the complexity of the PF with the maximum number of particles.

7 Experimental Results

For our experiments, we assumed all probability distributions to be Gaussian, and all sensors to have white

Table 1: Estimation errors averaged over 10 runs

No. of Factors	1	2	4
Mean	0.1143	0.1381	0.1968
(Standard Deviation)	(0.0360)	(0.0470)	(0.0314)

Table 2: Time taken for particle filter to complete estimation

No. of Factors	1	2	3	4
Time (s)	137.03	37.74	18.79	18.97

Gaussian noise with 0 mean and power 1 dbW. We estimated the state variables using the DBN factorings shown in Fig. 2(c) for 10 runs. Given m DBN-Fs, $D_i = \{\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i\}$, $i = 1, 2, \dots, m$, such that $\mathbf{X} = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \dots \cup \mathbf{X}_m$, for each run we computed the estimation error: $E = \frac{1}{|\mathbf{X}|} \sum_{X \in \mathbf{X}} \left(\frac{1}{T} \sum_{t=0}^T (X_t - X_t^{model})^2 \right)$, where T is the total simulation time, X_t denotes the estimated value of state X at time t , and X_t^{model} denotes the actual value of state X at time t obtained from the simulation model. Table 1 reports the mean and standard deviation of errors obtained from each factoring over all runs.

To demonstrate that the factoring scheme preserves the system dynamics, we hypothesized the difference in errors for the 2-factor and unfactored DBN would not be statistically significant, and the error for the 4-factor DBN would be significantly larger than the unfactored DBN. Further the difference in error for the 2-factor and 4-factor DBNs would also be statistically significant. We ran t -tests to establish significance of the differences. The tests for significance indicated that the errors obtained using the 2-factor DBN did not significantly differ from that obtained using the unfactored DBN ($p < 0.05$), while those obtained using the 4-factor DBN was significantly greater ($p < 0.05$). The test of significance between the 2- and 4-factor DBN showed that the error in the 4-factor DBN was significantly larger ($p < 0.05$). Therefore, the 2-factor DBN preserves dynamics of the unfactored DBN, whereas the 4-factor DBN, which has unobservable factors, does not.

Table 2 shows the average time taken by the slowest PF for each factoring to track system behavior for 1500 time steps. The time taken by a PF depends on the number of particles it uses. In our experiments, the number of particles used by a PF was proportional to the number of states in the DBN factor the PF was associated with. Hence, PF for unfactored DBN (with 1000 particles) took the most time, followed by the PF on the larger DBN-F of the 2-factor DBN (with 500 particles). The least amount of time was taken by the PFs applied to the 4-factor DBN, since its largest

DBN-F has 3 state variables, and hence, its PF used 300 particles.

8 Discussion and Conclusions

This paper presented an approach for factoring DBNs based on structural observability. Each of the DBN factors are conditionally independent from all other factors given the measurements that are communicated between them, thus preserving the dynamics of the global system behavior. Experimental results showed that factoring maintains inference accuracy in DBNs while improving the efficiency of DBN inference in the presence of sensor noise. Future work will focus on investigating stochastic notion of observability and its application to the design of DBN factors.

References

- Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing* 50(2), 174–188.
- Boyer, X. and D. Koller (1998). Tractable inference for complex stochastic processes. In *Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 33–42.
- Frogner, C. and A. Pfeffer (2008). Discovering weakly-interacting factors in a complex stochastic process. In J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), *Advances in Neural Information Processing Systems 20*, pp. 481–488. Cambridge, MA: MIT Press.
- Karnopp, D. C., D. L. Margolis, and R. C. Rosenberg (2000). *Systems Dynamics: Modeling and Simulation of Mechatronic Systems* (3rd ed.). New York, NY, USA: John Wiley & Sons, Inc.
- Koller, D. and U. Lerner (2001). Sampling in factored dynamic systems. In A. Doucet, N. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*. Springer.
- Lerner, U., R. Parr, D. Koller, and G. Biswas (2000). Bayesian fault detection and diagnosis in dynamic systems. In *Proc. of Seventeenth National Conference on Artificial Intelligence*, pp. 531–537.
- Mosterman, P. J. and G. Biswas (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE-SMCA* 29(6), 554–565.
- Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference, and Learning*. Ph. D. thesis, University of California, Berkeley.
- Mutambara, A. (1998). *Decentralized Estimation and Control of Multisensor Systems*. CRC Press.

- Ng, B. and L. Peshkin (2002). Factored particles for scalable monitoring. In *In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 370–377. Morgan Kaufmann.
- Samantaray, A. K. and B. O. Bouamama (2008). *Model-based Process Supervision: A Bond Graph Approach*. Springer.
- Sueur, C. and G. Dauphin-Tanguy (1989). Structural controllability/observability of linear systems represented by bond graphs. *Journal of The Franklin Institute* 326(6), 869–883.
- Sueur, C. and G. Dauphin-Tanguy (1991). Bond graph approach for structural analysis of MIMO linear systems. *Journal of the Franklin Institute* 328(1), 55–70.