# Estimating Order from the Causal Ordering

Jonathan Amsterdam*


## Abstract

Order (dimension of the state space) is a good complexity measure for dynamic models, and can be used in conjunction with a trace of dynamic behavior to determine model adequacy. I show how to use the causal ordering of a dynamic model to upper-bound the order of the entire model, and of a variable or variables in the model. I also show how to use the order of a variable to determine whether a model could plausibly account for a given dynamic behavior trace.


## 1 Introduction

Modeling is arguably the central task of physical reasoning. Without a model, prediction, analysis, and diagnosis are difficult or impossible. The modeling problem confronts us in many guises, but in the one I am concerned with, the modeler has access to behavior traces of the system to be modeled, perhaps in addition to other information about the system. This is a common situation. It arises when modeling existing systems whose outputs can be measured, and it occurs in design, where the behavior traces represent the desired behavior of the device. By a behavior trace I simply mean a plot of a quantity over time, an innocuous enough definition but one that carries the crucial assumption that dynamic (i.e. time-varying) behavior is of interest. Indeed, the central concept of this paper—model order—is important if and only if one is concerned with dynamic behavior.

Given a behavior trace, a natural question to ask is: what information can be extracted from the trace that will aid in modeling? In some important cases, part of the answer is that we can obtain the minimum order that a model must have in order to account for the behavior. This is a nice piece of information to have, for several reasons. First, it is non-parametric—it doesn't depend on the form of the potential models. In particular, it doesn't assume linearity. Second, it can often be obtained even from qualitative plots of behavior. Third, it provides the most important single fact about a model's complexity.

*This research was conducted while the author was at the MIT Laboratory for Artificial Intelligence. It has been supported in part by grant R01 LM 04493 from the National Library of Medicine.

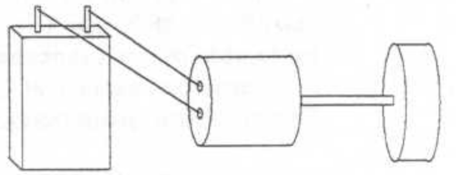

Figure 1: A battery, DC motor and flywheel

When dynamic behavior is the issue, the best measure of model complexity is order. The general rule is: the higher the order, the more complex the behavior. Other complexity measures, like number of equations, number of variables and so on, have no bearing whatever on dynamic behavior. Only order matters.

In this paper, I assume that the modeling process has produced a set of candidate models, and the task now is to compare each model to a given behavior trace to see if it can produce the behavior. (Alternatively, the modeler may interleave generation and testing.) Of course, simulation is one way to carry out this test of model adequacy, but simulation is expensive, it requires parameter-fitting in the numerical case, and may give erroneous answers in the qualitative case. The method I describe here is less general, but considerably cheaper and easier. That method is to estimate the order of the candidate model and to determine whether the order is high enough to generate the given behavior. The bulk of the paper describes how to estimate model order using a refinement of the well-known causal ordering technique. The estimate requires no algebra and can be done even if only qualitative information about the model is available. Near the end, I show how to use the estimated order to determine if the model can account for the behavior.

## 2 An Example

Consider an ordinary DC motor powered by a battery and attached to a flywheel (see Figure 1). When the battery is connected, the flywheel's speed might exhibit any one of several behaviors, three of which are shown in Figure 2. In Behavior 0, the flywheel comes up to speed instantaneously. In Behavior 1, the flywheel ramps up to its final speed. In Behavior 2, the flywheel's velocity overshoots its final value before settling down.

These three behaviors suggest three quite different sorts of models for the system. The difference between

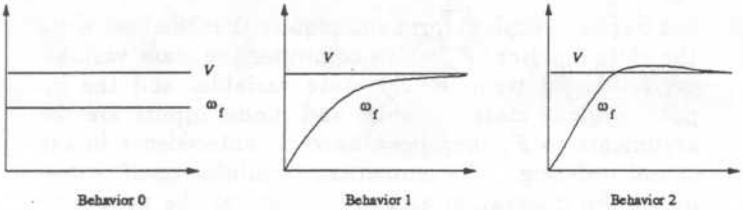


Figure 2: Three possible behaviors of the motor-flywheel system. $V$ is battery voltage, $\omega_f$ is flywheel velocity.

| | |
|---|---|
| $V = c$ | battery supplies constant voltage |
| $\omega_m = f_1^+(v_m)$ | motor shaft velocity $\propto$ motor voltage |
| $\tau_m = f_2^+(i_m)$ | motor shaft torque $\propto$ motor current |
| $\tau_f = J_f\dot{\omega}_f$ | flywheel torque vs. angular acceleration |
| $V = v_m$ | battery connected to motor leads |
| $\omega_m = \omega_f$ | motor shaft connected to flywheel |
| $\tau_m = \tau_f$ | motor shaft connected to flywheel |

Figure 3: Model 0 of motor-flywheel system

the first and the others is most significant. Behavior 0 has no time-dependent element; the velocity is (as far as we can tell from the given information) a function solely of the input voltage and not, as with the other two behaviors, a function of both input and time. The difference between Behaviors 1 and 2 is a little more subtle; I will return to it later.

Figures 3 and 4 present two models (henceforth Model 0 and Model 1) of the motor-flywheel system. (Monotonically increasing functions are written as $f^+$.) They differ only in that Model 0 treats the battery as an ideal voltage source, while Model 1 treats it as having resistance, which will result in its voltage falling off for high loads. For both models, there is a path in the causal ordering from the input voltage $V$ to the flywheel speed $\omega_f$, so one might say (as Nayak [7] does) that in both models $V$ "causes" $\omega_f$. Yet the models predict different dynamic behaviors. In particular, Model 0 predicts Behavior 0 and Model 1 predicts Behavior 1.

In what follows, I explain how to use the causal ordering to distinguish between the two models, based on information in the observed behavior of the system, without resorting to simulation or algebraic manipulation. The method can also be used to determine that neither model is capable of accounting for Behavior 2.

| | |
|---|---|
| $V = c$ | battery nominal voltage is constant |
| $v_r = f_3^+(i_r)$ | battery internal resistance |
| $v_b = V - v_r$ | battery output voltage reduced by resistance |
| $\omega_m = f_1^+(v_m)$ | motor shaft velocity $\propto$ motor voltage |
| $\tau_m = f_2^+(i_m)$ | motor shaft torque $\propto$ motor current |
| $\tau_f = J_f\dot{\omega}_f$ | flywheel torque vs. angular acceleration |
| $v_b = v_m$ | battery connected to motor leads |
| $i_r = i_m$ | battery connected to motor leads |
| $\omega_m = \omega_f$ | motor shaft connected to flywheel |
| $\tau_m = \tau_f$ | motor shaft connected to flywheel |

Figure 4: Model 1 of motor-flywheel system

# 3 Order

If we consider models that consist of a finite set of ordinary differential equations, then at any point in time, a *state*—a finite set of values—suffices to describe all that is relevant about the model. The entire future and past of the model's behavior can be predicted from its state. The *order* of a model is the number of values needed to describe its state. These values are called *state variables*. Typically (and hopefully) one can write the set of equations in the form

$$\dot{x} = F(x, u(t))$$

where $x$ is the *state vector,* the vector of state variables, and $u(t)$ is the system input. This form, which I will call the *explicit form,* admits of relatively easy computer simulation using a variety of techniques, and also provides one motivation for the causal ordering process I will discuss in the next section.

One way to find the order of model is to place its equations in explicit form and determine the size of the state vector. But the symbolic math required to do this may be difficult or time-consuming, or the full equations may not be available—the model might be described qualitatively. If an energy-based modeling framework is adopted (see, e.g. [5]) and the energy-storage elements of the model can be found, then a quick upper bound on the order can be obtained, because the order cannot be greater than the number of energy-storage elements. In general, we can obtain a set of candidate state variables directly from the equations, and the size of this set is an upper bound on the order. But we can improve that upper bound in two ways.

First, we can get a better upper bound by trying to find dependencies among the candidate state variables. We know that the order of Model 0 is at most one, because it has only one energy-storage element (the flywheel, an inertia). In fact, however, its order is zero. The order of Model 1 is indeed one, but if we added a second flywheel next to the first on the motor's rigid shaft, we would still have a first-order model despite having two inertias. We can detect both of these problems, and others, without doing any algebra.

A second improvement is to ask (and answer) a more focused question: we would like, not the order of the entire model, but only that of the variable of interest.[1] The model order is the amount of state required to determine all the model's variables throughout time, but some of those variables might require less state to predict. For instance, if an ideal voltage source (like the battery of Model 0) were driving two loads in parallel, then the or-

[1] I assume throughout that the output variable of interest is one of the variables in the model's equations. If not, the model is patently inadequate.

der of the overall model might be higher than necessary to predict the complete behavior of one of the loads.

# 4 Causal Ordering

The causal ordering of a set of equations is a graph of the computational dependencies between variables in the equations. The graph's nodes are the variables, and there is a directed arc from $v_1$ to $v_2$ if $v_1$ is used to compute $v_2$.

In [3], causal ordering is defined in terms of subsets of solvable equations. Nayak [7] defines it as a 1-1 function from equations to the variables they casually determine. I prefer the constraint-propagation method of [2], which brings out the computational nature of causal ordering.

Begin with constants and variables whose values are determined exogenously (e.g. inputs to the model). If there is an equation all but one of whose values are among these known quantities, and if the unknown can be computed from the knowns, then draw arcs from each known to the unknown, and add the unknown to the set of knowns. Continue in this fashion. If none of the remaining equations are usable because they each contain more than one unknown, choose ("plunk" in deKleer and Brown's terminology) a variable, assume it is known and continue constructing the graph. Eventually, this chosen variable should be determined by some other equation, forming a cycle in the graph; this cycle indicates a set of simultaneous equations.

An important constraint on the process is that each variable can be determined by only one equation. If this constraint is violated, the causal ordering is inconsistent. If variables have been plunked in order to break cycles, a consistent ordering may still be possible by making different choices. Backtracking or some other search technique is necessary here.

For dynamic models, assume equations of the form $\dot{v} = \frac{d}{dt}v$ for each variable whose derivative also occurs in the model. We will call the arc arising from this equation a *d-arc*. In a proper causal ordering, the equation should be ordered from $\dot{v}$ to $v$; the resulting arc is called an *integration arc*. I will also call an integration arc an *integral d-arc;* in the other direction it is called a *derivative d-arc*.

If all equations are used and some non-exogenous variables are still undetermined (have no incoming arcs), then the model is underspecified: the equations are insufficient to determine all the variables. Otherwise, the causal ordering is *complete*.

If a complete causal ordering has no cycles, then it is easy to solve for all the variables by repeated substitution. A cycle with no integration arcs indicates a set of simultaneous algebraic equations. There is no guarantee that they can be solved to obtain unique values for all their variables. But for the rest of the paper, we will assume that they can be solved.

Here is the relationship among causal ordering, the explicit form $\dot{x} = F(x, u(t))$, computation, and numerical simulation. Numerical simulators assume the equations can be put in explicit form and require that the user write the *state function*, $F$, which computes the state variable derivatives in terms of the state variables and the input. Because state variables and model inputs are the arguments to $F$, they need have no antecedents in the causal ordering. The numerical simulator itself implements the integration arcs, supplying, at the next time step, new values of the state variables computed using their derivatives.

# 5 Estimating Model Order

Estimating the order of a variable of interest is a two-step process. First, we upper-bound the model order by finding dependencies among candidate state variables. Then we use the causal ordering to determine the variable's order. This section discusses the first step. First some definitions.

**Definition 1** *A variable is a* root *in a causal ordering if it has no incoming arcs, except possibly an integration arc.*

**Definition 2** *The* algebraic support *of a variable $v$ in a causal ordering is the set of roots $r$ such that there is a path from $r$ to $v$ that contains no d-arcs.*

The algebraic support of a root is the singleton set containing itself.

**Definition 3** *A variable is* dependent *if its algebraic support does not contain itself.*

If all algebraic cycles are solvable, a dependent variable can be computed algebraically from other given quantities.

**Definition 4** *A variable is a* candidate state variable *if both it and its derivative appear in the model equations.*

## 5.1 Eliminating Candidate State Variables

Some candidate state variables may not, in fact, be state variables. There is a two-step procedure for finding these impostors. The first step has a clear intuition: we will try to detect candidate state variables that can be computed from other variables.

Begin by constructing a causal ordering for the model, starting from the inputs and the model parameters. When and if a variable needs to be plunked, choose a derivative of a candidate state variable if any remain undetermined. Note that this will force the d-arc of the variable to be integral. The intuition behind determining candidate state variables by integration comes from the fact that state variables as well as model inputs are included in the inputs to the state function.

When the complete causal ordering has been constructed, look for candidate state variables whose d-arcs point in the derivative direction (from the variable to its derivative). Such a candidate state variable is dependent: it must have algebraic support, because it must have an incoming arc; and that support cannot include the variable itself, because if such a cycle were present, it would have been broken by plunking the variable's derivative, which would have resulted in an integral d-arc. Any dependent candidate state variables are not, in fact, state variables. They can be computed from other quantities, so they contribute nothing to the computation of the model's state.

### 5.1.1 Example

Before proceeding to the second step, let us apply the first step to Model 0 to show that it is zeroth-order. Our candidate state variable is the flywheel's angular velocity, $\omega_f$. We begin with the constant voltage $V$. It determines $v_m$ from the equation $V = v_m$. Then $v_m$ determines $\omega_m$, which determines $\omega_f$, which determines $\dot{\omega}_f$.

We can stop here; our lone candidate state variable has a derivative d-arc, so it is dependent. Thus the model has no state variables; it is zeroth-order.

In this simple example it is easy enough to do the algebra and obtain $\omega_f = f_1^+(V)$, showing that the flywheel's angular velocity is is function of the input voltage. But in general the algebra may be harder than computing the causal ordering, or there may be no closed-form algebraic solution, as might arise if one were confronted with a set of simultaneous nonlinear equations, or there may be insufficient information to do the algebra, as when the model is qualitative.

### 5.1.2 The Second Step

With some additional work, we can eliminate more candidate state variables. The idea is to look for dependencies among derivatives of the candidate state variables. If for candidate state variables $y, x_1, x_2, \ldots, x_n$ we can show that

$$\dot{y} = \sum_{i=1}^{n} k_i \dot{x}_i \tag{1}$$

where the $k_i$ are constant, then

$$y = \sum_{i=1}^{n} k_i x_i + c$$

so $y$ is computable from the $x_i$ and thus is not needed in the state vector. However, an initial condition for $y$ is still required, in order to determine the constant of integration $c$.

The procedure begins analogously to that of the first step, except that we prefer plunking the candidate state variables, rather than their derivatives; this forces their d-arcs to point in the derivative direction. (Before performing this step, remove the dependent variables found in the first step from the set of candidate state variables.) When the causal ordering is complete, it is the integral d-arcs that indicate potentially dependent candidate state variables. Determine, via algebra, if the dependency is in the form of Equation 1. If so, remove the variable from the set of candidate state variables.

The remaining candidate state variables are an acceptable set of state variables that will allow the model to be expressed in explicit form, and the size of the set is an upper bound on the order of the model. But it is still only an upper bound. I know of no substitute for doing the algebra if we desire the exact order. As a trivial example of how the causal ordering method is sensitive to algebraic form, consider $y = x - x$. Here, though $y$ is constant, a causal ordering might have an arc from $x$ to $y$. Of course, less obvious examples involving multiple equations are not hard to come by, and do arise in practice.

In one case, we can be sure that the bound is tight: when it is zero. This is also arguably the most important case, because of the vastly different analysis techniques for algebraic versus dynamic models.

## 6 The Order of Variables

Now that we have identified the model's state variables, we are in a position to determine the order of a particular variable. As I mentioned earlier, this may be less than the order of the entire model.

First it is necessary to define another kind of support.

**Definition 5** *The* full support *of a variable $v$ in a causal ordering is the set of roots $r$ such that there is a path from $r$ to $v$.*

Full support is just like algebraic support, except that d-arcs are included.

To find the order of a variable in a model, construct a complete causal ordering for the model, preferring derivatives of state variables when plunking is required. (This construction is like that of Section 5.1.) *The order of a variable is the number of state variables in its full support.* The algebraic support won't do: that would give us the state variables needed to compute our variable of interest $v$ given the current state-variable values; in other words, it would tell us what we need to compute $v$ for one invocation of the state function $F$. The full support includes the state variables necessary to compute, not only $v$, but also those state variables' derivatives as well. In effect, we have isolated a subset of our original model sufficient to predict $v$ for all time.

The order of a set of variables is the number of state variables in the union of their full supports.

## 6.1 Applications

I showed above that Model 0 is zeroth-order, so all its variables are. The method of Section 5 determines that Model 1 is first-order, and doing the algebra confirms that. From its causal ordering, shown in Figure 5, it is evident that every variable except for $V$, the nominal battery voltage, contains the state variable $\omega_f$ in its full support. Thus except for $V$, the variables of Model 1 are first-order.

Now say we added the angular position $\alpha_f$ of the flywheel to Model 1. This is another state variable, and the causal ordering of the resulting model is shown in Figure 6. The model is now second-order because there are

$$
\begin{array}{ccccccccccc}
V & \rightarrow & v_r & \rightarrow & i_r & \rightarrow & i_m & \rightarrow & \tau_m & \rightarrow & \tau_f \\
 & & \uparrow & & & & & & & & \downarrow \\
 & & v_b & \leftarrow & v_m & \leftarrow & \omega_m & \leftarrow & \boxed{\omega_f} & \Leftarrow & \dot{\omega}_f
\end{array}
$$

Figure 5: Causal ordering for Model 1 (d-arc shown with double arrow, state variables boxed)

$$
\begin{array}{ccccccccccc}
V & \rightarrow & v_r & \rightarrow & i_r & \rightarrow & i_m & \rightarrow & \tau_m & \rightarrow & \tau_f \\
 & & \uparrow & & & & & & & & \downarrow \\
 & & v_b & \leftarrow & v_m & \leftarrow & \omega_m & \leftarrow & \boxed{\omega_f} & \Leftarrow & \dot{\omega}_f \\
 & & & & & & & & \downarrow & & \\
 & & & & & & & & \boxed{\alpha_f} & \Leftarrow & \dot{\alpha}_f
\end{array}
$$

Figure 6: Causal ordering for Model 1 with flywheel position added

two state variables, but all variables except for $\alpha_f$ and $V$ are first-order.

# 7 Using Order for Modeling

Order is a good measure of complexity for dynamic models, because it corresponds well to the complexity and variety of behaviors a system can exhibit. The number of model variables, or components, or processes, means nothing when it comes to dynamic behavior; only the number of state variables is important.

As I outlined at the beginning, order can be also used to determine whether a model has the potential to adequately describe a dynamic behavior.

The test for whether a zeroth-order model is adequate is simple. Given a trace over time of an output and one or more inputs, plot the output against the inputs. If the result is a function, a model in which the output variable is zeroth-order is indicated. If not, the system exhibits dynamic behavior, and the output variable must be at least first-order. This criterion admits Model 0 as a possible model of Behavior 0, but not of Behaviors 1 or 2, because in these latter the single input value corresponds to many output values.

Higher-order tests use the phase-space non-intersection constraint, which says that a phase-space trajectory cannot intersect itself (except to form a cycle). The general test for order $n$ requires at least $n$ output variables. Say there are $k$ inputs. Plot the trajectory of the $(n+k)$-tuple of output variable and input values at successive points in time. An intersection implies that the order of the set of output variables exceeds $n$.

An important special case is the first-order test for autonomous models. Say there are no inputs, or they are all constant. Then if the output variable of the behavior is (non-strictly) monotonic, either increasing or decreasing, a model in which that variable is first-order could account for the behavior. If the behavior ever changes direction, the output variable must be at least second order. The proof is simple: a first-order phase space is just a line, and if a trajectory changes direction along that line, it will intersect itself. Hence a change in direction implies a model order greater than one.

The first-order test rules out both Model 0 and Model 1 for Behavior 2. The input in Behavior 2 is constant, so the test is applicable, and the output is non-monotonic, so it must be at least a second-order variable. Note that our inability to find the exact order for the models is no hindrance here; the upper bounds suffice.

# 8 Related Work and Conclusion

Causal ordering has found a variety of uses in the qualitative reasoning community. Iwasaki [3] extended Simon's [10] original conception of causal ordering to dynamic and mixed systems. Williams [11] used causal ordering to remove interactions superfluous to a query. Nayak [7] used causality to formulate a modeling standard, a criterion that an adequate model must meet. For a model to have a shot at answering the question "Does quantity $q_1$ cause quantity $q_2$?" its causal ordering must have a path from $q_1$ to $q_2$. My method can be view as an extension to Nayak's. He can determine whether a model predicts that one quantity causes another; I add information about the dynamics of the quantities' causal connection.

The method for finding dependent state variables that I presented in Section 5 is based on the bond graph causality assignment procedure, first presented in [8]. Bond graph texts such as [5] present the first step only; the second step appears first in [4]. The full procedure is codified in [9]. Integral and derivative d-arcs correspond to integral and derivative causality on energy-storage elements. Bond graph causality uses a constraint that I have not captured; namely, asserting that a flow is determined in one direction along bond implies that an effort is determined in the other direction. This constraint arises from the energy-based semantics of bond graphs. The ramifications of ignoring the constraint are not yet clear to me. My method is more convenient in that it does not require the model to be described as a bond graph (but there are many advantages, not discussed here, to using bond graphs for modeling physical systems; see [1]).

Besides generalizing the bond graph causality technique, I have also shown how model order can aid in finding an adequate model when traces of system behavior are available. The idea is to estimate the order of the output variable, and try to show whether the order is high enough to produce the given dynamic behavior. There are other methods for estimating order from behavior [6], but they assume linearity or a specific model structure.

There is more to say about the interaction of order and behavior. In future work, I hope to show how to use the causal ordering to identify order-increasing modifications for models whose order is too low to account for a behavior.

## Acknowledgments

Neville Hogan taught me bond graphs. Pandurang Nayak helped me understand his work. Peter Szolovits provided welcome support and encouragement.

## References

[1] Jonathan Amsterdam. *Automated Qualitative Modeling of Dynamic Physical Systems.* PhD thesis, MIT, 1993.

[2] Johan de Kleer and John Seely Brown. Theories of causal ordering. In Daniel S. Weld and Johan de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, San Mateo, CA, 1990.

[3] Yumi Iwasaki. Model based reasoning of device behavior with causal ordering. Technical Report CMU-CS-88-172, CMU, August 1988.

[4] Dean Karnopp. Alternative bond graph causal patterns and equation formulations for dynamic systems. *Journal of Dynamic Systems, Measurement, and Control*, 105:58–63, 1983.

[5] Dean Karnopp and Ronald Rosenberg. *System Dynamics: A Unified Approach.* John Wiley & Sons, New York, 1975.

[6] Lennart Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, N.J., 1987.

[7] P. Pandurang Nayak. Causal approximations. In *Proceedings of the AAAI*, 1992.

[8] Henry Paynter. *Analysis and Design of Engineering Systems.* Unpublished, 1961.

[9] R. C. Rosenberg and J. Beaman. Clarifying energy storage field structure in dynamic systems. In *Proceedings of the American Control Conference*, 1987.

[10] Herbert A. Simon. On the definition of the causal relation. *Journal of Philosophy*, 49:517–528, 1952.

[11] Brian C. Williams. Capturing how things work: Constructing critical abstractions of local interactions. In *Workshop on the Automatic Generation of Approximations and Abstractions*, 1990.