Understanding model behavior using loops that matter.
William Schoenberg\textsuperscript{1,2}, Pål Davidsen\textsuperscript{1}, Robert Eberlein\textsuperscript{2}

\textsuperscript{1}University of Bergen, Norway
\textsuperscript{2}isee systems inc. Lebanon, New Hampshire USA
Abstract
The link between structure and behavior is central to System Dynamics, but effective tools for understanding that relationship still elude us. The current state of the art in the field of loop dominance analysis relies on either practitioner intuition and experience or complex algorithmic manipulation in the form of eigenvalue analysis or pathway participation metrics. This paper presents a new and distinct method to find the 'loops that matter' in generating model behavior. This is a numeric method capable of determining the impact for every loop in a model and identifying which dominate behavior at each point in time. The method was inspired by observations on variable value changes during simulations and has been refined using empirical evaluation on a variety of different models. In addition to explaining behavior, the method shows promise for improving visualization and aggregation of simulation results.

The problem
The strong link between structure and behavior is fundamental to system dynamics (Sterman, 2000). A model uses parametric and structural assumptions to derive behavior. A system dynamics practitioner at a general level must go through the following process: create a structure representative of the problem under consideration, understand how that structure works, and figure out how to improve that structure to address the problem. The second step in that process is the focus of this paper and is the key to successfully performing the third step. By referring back to the model structure, the practitioner can explain the reasons why an observed behavior has been produced (Richardson, 1996). Based on that understanding, the practitioner may also propose changes in input values (response sensitivities) or model structure (system configuration) that will cause a more favorable behavior to be produced.

The current state of the art in the field relies on either practitioner intuition and experience (the art of modeling and model analysis) or complex algorithmic analysis. The former is taught as part of the methodology of model building, while the latter comes from 40 years of work on techniques to derive and explain model behavior based on the analysis of structure (see for example: Graham, 1977; Forrester 1982; Eberlein, 1984; Davidsen, 1991; Mojtahedzadeh, 1996; Ford, 1999; Saleh, 2002; Goncalves, Mojtahedzadeh et al., 2004; 2009; Saleh et al., 2010, Kampmann, 2012; Hayward and Boswell, 2014; Moxnes and Davidsen, 2016; Oliva, 2016 and Hayward and Roach, 2018.

Ford (1999, p.4-5) clearly stated the needs of the system dynamics field as they apply to loop dominance analysis:

“To rigorously analyze loop dominance in all but small and simple models and effectively apply analysis results, system dynamicists need at least two things: (1) automated analysis tools applicable to models with many loops and (2) a clear and unambiguous understanding of loop dominance and how it impacts system behavior.”
In this paper we discuss a new loop dominance analysis method and demonstrate its performance on the two tests set out by Ford. For the purposes of this paper, we define one loop as dominant over the others if it is responsible for generating at least 50% of the changes in behavior observed in the stocks of the model at a specific point in time. This measure is relative across feedback loops in the system. Often, there will be one singular dominant loop at each point in time for the model, but there are cases where no single loop will be solely dominant and we therefore must consider multiple loops together as being dominant. This definition holds true for the cases where all stocks are connected to each other and themselves by the network of feedback loops in the model either directly or indirectly. For models where there are stocks which do not share feedback loop relationships, we consider each subcomponent of inter-related feedback loops separately and we refer to each partition of the model’s structure along those lines as having a separate behavior origin. An in-depth definition of this concept is given and explained in the ‘Defining loop scores’ section of this paper.

The purpose of this research is to create an analysis method that is both correct and practical so that it will be used. While this method may not generate new insights relative to existing analysis methods, it has two characteristics that make it more practical. First, it is relatively easy to understand for anyone who has built and analyzed system dynamics models, even those with a limited mathematical background. Second, it lends itself completely to presenting behavior over time graphs that mix loop dominance measures and model behavior making it easy to see the evolution of which loops matter. These characteristics, combined with the very straightforward computational techniques we employ, mean that such analysis can easily be built into existing system dynamics modeling environments. This will make looking at which loops matter as easy as it is to look at a graph of behavior. Such ease of use means the methods will regularly, if not universally, remove one more obstacle in our ability to do good work.

Literature review
The current state of the art in the use of mathematical methods for determining loop dominance revolves around two methods. The first one is based on eigenvalue elasticity analysis, the second one, uses the pathway participation metric and causal pathways.

Eigenvalues and eigenvectors, specifically eigenvalue elasticity analysis (EEA)
Forrester (1982) was the first to document that eigenvalue elasticities could be used to explain the relative contributions of different loops in models of linear systems. Since then, the formal method of eigenvalue elasticity analysis (EEA) has been further developed and is used to determine how model structure produces the dynamic modes of behavior for the model, specifically those characterizing the state variables or stocks, (Saleh, 2002), (Kampmann et al., 2006), (Saleh et al., 2010), (Oliva, 2016). Using EEA, the structure of a model is characterized by the eigenvalues and eigenvectors of that model. This analysis is built on the observation that the behavior of a linear system can be expressed as a linear combination of behavior modes, each characterized by a decoupled (or pairwise coupled) set of eigenstates. (Saleh et. al., 2010). EEA is applied to examine both link and loop significance with regard to the dynamic behavior of the model. It does so by identifying the relationship, expressed in the form of the elasticity, between the parameters that make up the gains of an individual feedback loops in the model
and the eigenvalues (and sometimes eigenvectors) that characterizes the dynamic behavior of
the model. The significance of a loop to behavior is expressed by the eigenvalue elasticity of its
gain, that is how strongly a change in the gain impacts the eigenvalue associated with the
behavior of interest. Note that this may not only be used to identify the root cause of a model’s
behavior, but also the leverage points for controlling the system (policy entry points) if the
model is an accurate representation of the system. Kampmann (2012) developed the concept
of the independent loop set (ILS) which filters all of the loops in the model into a singular set of
independent loops which represent the full behavior of the model so that the analysis can be
effectively completed and interpreted. Oliva (2004) extended Kampmann’s work on the ILS by
developing an ILS composed only of geodetic loops which he termed the shortest independent
loop set (SILS) which is the de-facto standard for determining which loops to analyze in EEA.

The purpose of EEA is more encompassing than the other methods discussed in this paper
(loops that matter, or the pathway participation metric). EEA is a general method which
describes the behavior state space of the model and speaks not only to what behavior the
model is producing with a single set of input values, but what behavior modes are capable of
being produced using any set of input values. According to Oliva (2016) the EEA approach
satisfies Checkland and Scholes (1990) three E’s criteria to assess performance. It is efficacious,
efficient, and effective.

The downsides of the EEA method are that it is mathematically complex, requires a deep
understanding of linear algebra, and may be applied effectively to only a very small subset of
models unless they are modified (Saleh et al., 2010). Specifically, models must be linearized to
make them well suited for such an analysis, a process that is hard to automate (though that is a
problem actively being worked on). Additionally linearization may change the simulation
results. Oliva (2016) when analyzing his service quality model had to, among other changes,
remove a stock in order to produce a full rank system matrix which was necessary to perform
an EEA analysis and change model equations to ensure the model was continuously
differentiable. Those changes did have an impact on the simulation results.

Pathway participation metric (PPM) and other causal pathway techniques
The pathway participation metric (PPM) approach does not use eigenvalues to describe model
structure. Rather, it focuses on the links between variables (Mojtahedzadeh et al, 2004). The
starting point in the PPM approach is the behavior of a single variable, typically a stock. The
behavior of that single variable is partitioned in time, based on periods in which the variable
maintains slope and convexity(first and second time derivatives not changing sign)
(Mojtahedzadeh et al, 2004). This then limits the behavior of the variable at each of these
phases to 7 patterns enumerated by Mojtahedzadeh et al, (2004). The PPM approach then
determines dominance by tracing along the causal pathways between the stock under study
and its ancestor stocks to determine which structure is most influential in explaining the
pattern of behavior exhibited by that stock during the selected phase. Mojtahedzadeh et al.,
(2004) explains that it does this by determining the magnitude of the change in the net flow of
the stock under study by making minute changes to that stock. The method then compares
these changes in the net flow to determine the change with the largest magnitude in the same
direction as the stock under study thereby identifying the most important (dominant) pathway governing the behavior of that stock during that phase.

Relative to the general EEA that yields results covering the entire behavior space (all modes of behavior that may potentially be produced by the model structure), PPM is considerably more specific. PPM relies on the state values in the model during each phase of analysis and is unable to provide information about behavior that has not manifested. Using this method, therefore, we can only determine the impact of causal pathways based on the given set of values for inputs and parameters. The only way to determine what behavior modes a model is capable of producing using PPM is to specifically generate each of them (potentially via a Monte Carlo sensitivity analysis) and analyze each one independently.

Among the benefits of the PPM approach, relative to the EEA approach is that, for it to work, it does not require manipulation of the structure of the model, nor is there a need for linearization. Moreover, according to the research by Mojtahedzadeh (1996) application of the PPM method will cause a convergence on a unique piece of structure as the one most influential with regard to the behavior phase under study. Kampmann and Oliva (2009) state that one of the key benefits to the PPM method is its direct connection between behavior and structure.

Kampmann and Oliva (2009) have criticized PPM for its inability to clearly explain oscillatory systems and also because PPM can fail to identify structure when there are two pathways of similar importance (Kampmann and Oliva, 2009). Hayward and Boswell (2014) have responded to those criticisms by simplifying PPM via the loop impact method. The loop impact method can be implemented in a standard system dynamics model (and software) without any change in the underlying software by adding equations to the model. The key differences of the loop impact method as compared to PPM is that it does not look for dominant pathways, but instead focuses on the direct impacts that one stock has on another (Hayward and Boswell 2014). In addition, the loop impact method identifies instances where multiple loops are required to explain the behavior of a stock.

Expanding on the work done by Hayward and Boswell (2014), Hayward and Roach (2018) have developed a framework around the loop impact method couched in the mathematics of Newtonian physics to explain the model as a series of interacting forces. The stated purpose of the underlying common research thread between them, the loop impact method, is to provide a more intuitive and complete understanding of loop dominance in system dynamics models.

**The loops that matter method**

In this paper, we present the LTM (Loops That Matter) method which determines loop dominance for models of any size, complexity, or dimensionality. Like PPM this method is derived, in part, from observations of the way that experienced modelers perform analysis to determine the sources of observer behavior. Just as when analysis is done by experimentation and examination, all of the things we look at in measuring impact are part of the real model
structure. This has the advantage of making it easier to interpret the different components of impact, but the disadvantage of not having a canonical form (eigenstate) that rigorously defines the simplest abstract system from which behavior can derive. Given the lack of canonical form, there are some decisions around computational details that are not fully defined, and we have settled on those that give good results for a variety of test cases. Though not unique, the approach we described is, as discussed below, invariant to the addition of intermediate structure and has proven to be both simple and remarkably informative for a variety of models and behavior modes.

LTM is computed using values realized during a simulation and can therefore be used on continuous and discontinuous models without requiring linearization. We compute loop scores which are normalized values representing the polarity and strength of each feedback loop at each time in the simulation. The analysis of the relative loop scores at a particular point in time identifies the loops that are dominating behavior. The display of the loop scores over time builds understanding of why the model behaves the way it does. Due to their design, loop scores are completely insensitive to the number of variables and links in a loop. Assuming the same model structure, there can be many variables with simple equations, or a small number of variables with complex equations and the values for the loop scores of those models will be the same.

We use the standard definition of a loop as a set of interconnections between variables in a model that form a closed path from a variable back to itself. The interconnections we refer to as links. Loop scores are computed as products of link scores, and the definition of a link score is tailored to this specific use. The link score computation as described in equations 1 and 2 has been designed for the purpose of determining loop dominance, specifically to be used in the loop power and loop score calculations described in equations 3a and 3b. As will be shown in this section of the paper link scores are not a general metric to describe the strength or importance of any specific link in isolation of the loops it is a member of. The most obvious manifestation of the link score’s lack of generality is that a link from an unchanging variable, such as a constant (or even a variable which is temporarily constant), has a score that is definitionally 0 over the time periods where the variable is constant. This is so because when the links in loops do not change, the loop is inactive and therefore not currently of consequence. This does not mean that such parameters are unimportant to the loop dominance analysis of a model. Even though the link score for all links from parameters to variables are 0, the parameters define the equations and thus the link and loop scores for the variables in which they are used. We address this in the discussion of the inventory workforce model.

We compute link scores for the influence of flows on stocks over time as well as the direct (instantaneous) algebraic influence of a flow or auxiliary variable on another. Conceptually these will be treated the same, both being multiplied when calculating a loop score, but they do require a different computation as discussed below.

**Defining link scores for auxiliary variables**
To simplify the presentation, we will define the link score assuming there are two inputs (x and y) to the dependent variable z characterized by the equation 
\[ z = f(x, y) \].

This easily generalizes to the case where there are more (or fewer) inputs into the equation for z.

The link score for the link \( x \rightarrow z \) is:

\[
LS(x \rightarrow z) = \begin{cases} 
\left( \frac{\Delta x}{\Delta z} \cdot \text{sign} \left( \frac{\Delta z}{\Delta x} \right) \right), & \Delta z = 0 \text{ or } \Delta x = 0 \\
0, & \Delta z = 0 \text{ or } \Delta x = 0 
\end{cases}
\]  

(1)

Where \( \Delta z \) is the change in z from the previous time to the current time. \( \Delta x \) is the change in x. \( \Delta x/z \) is the change in z with respect to x. From a computational perspective \( \Delta x/z \) is the amount z would have changed, conditionally, if x had changed the amount it did, but y had not changed. The first major term in this equation represents the magnitude of the link score, the second is the link score polarity.

The exceptions for no change in x or z are included for completeness, but are not important to the end goal of calculating loop dominance. In the case of no change in x, the link score magnitude goes to 0 and the sign, though not computable, is not relevant as positive and negative zero are equivalent for this exercise, both yielding an inactive link and therefore inactive loop. In the case of no change in z, any link using z as an input will not be active because no change will be attributed to z in any equation that z appears as an input to. Therefore any loops involving z are inactive and definitionally not dominant so we defined the value of the link score as 0 in this case.

The first major term in the link score equation is the link score magnitude \( \left| \frac{\Delta x/z}{\Delta z} \right| \) which describes the effect (force is a good analogy) that an input x has on an output z, and is relative to all of the effects on z. Unlike a partial derivative, which describes how sensitive z is to changes in x, the magnitude describes how much the change in x changed z. If the change in x is very small, this magnitude will be very small. It is a dimensionless quantity and, if all of the effects are in the same direction, it is the fraction of the change in z that results from the change in x. We refer to this measure as a contribution from x to z or as the strength of the link from x to z. If the formulation of z is linear, then the values are restricted to the range between 0 and 1.

When there are opposing forces in a non-linear equation the link score magnitude may have a very large value, but this does not harm the overall analysis of loop dominance because it is the relative values for loop strength that are analyzed at each point in time and large magnitudes balance each other in this comparison.

The second major term in the link score equation is the polarity of the link \( \text{sign} \left( \frac{\Delta x/z}{\Delta x} \right) \) which is defined as the sign of the partial difference at time t. This formulation is functionally the same as the one used in Richardson 1995. We use the partial difference notation in order to
maintain consistency with the link score magnitude whereas Richardson uses the partial derivative notation. Our reformulation of Richardson’s polarity makes it easier to calculate the link score because the \( \Delta z \) value can be re-used in both terms. Additionally, if the equations are discontinuous, a partial derivative might have a different sign from the realized impact making it more difficult to properly assign polarity.

**Defining link scores for stocks**

Because stocks represent an integration process where they change as a result of flows, not because of changes in flows, the computation of link scores for links going into stocks is different. Assume the stock equation \( s = \int (i - o) \) where \( s \) is the stock, \( i \) is the inflow, and \( o \) is the outflow. We assume a single inflow and outflow for simplicity of presentation, the generalization to multiple inflows and outflows is straightforward.

\[
\begin{align*}
\text{Inflow: } & LS(i \to s) = \left( \left| \frac{i}{i - o} \right| \right) * 1 \\
\text{Outflow: } & LS(o \to s) = \left( \left| \frac{o}{i - o} \right| \right) * -1
\end{align*}
\]

We use the same form as we do in previous link score for clarity, and again assume that the link score is 0 for all links from a flow to a stock if the net flow of the stock \( (i - o) \) is 0. As discussed above, the assumption of a 0 link score in this case does not change any loop scores since the links coming out of the stock will have a 0 score.

In this formulation a value of 0 for an inflow or outflow will result in a 0 link score. If the inflow and outflow are nearly balanced, so that the stock there is only a small change in the stock, the link scores for the links from the flows to the stock will be large, but close in value. This happens because the denominator of the equations (2) approaches zero faster then the numerators in such cases.

For models where inflows and outflows are not explicit, but implicitly represented by an equation such as \( avg = \int ((input - avg)/st) \) (for example a smooth, which calculates a smoothed average \( avg \) based on an \( input \) and smoothing time \( st \)), we decompose the expression into an explicit net flow such as \( ((input - avg)/st) \). The link score for this new expression then ends up being the link score that matters, since the stock portion is 1 by definition.

**Link score computation examples**

Table 1 demonstrates the process for calculating the link score magnitude for an auxiliary (non-stock) variable. It uses the equation \( z = x + y \) to demonstrate how to calculate a link score magnitude. In this specific case there are two link score magnitudes that must be calculated, one for the link \( x \to z \) and one for the link \( y \to z \). To calculate the link score magnitude for the link \( x \to z \) first determine \( \Delta z \) which is the actual change in \( z \) (3). Next determine the change in \( z \) with respect to \( x \) which is represented with the symbol \( \Delta x \) by substituting in to the equation for \( z \) the previous value of \( y \) (4) and the current value of \( x \) (7). Then take the computed value of \( z \) using those values (11) and subtract from it the previous value of \( z \) (9) to yield \( \Delta x z \) (2). To
complete the calculation of the link score magnitude divide \( \Delta_x z \) (2) by \( \Delta z \) (3) to get the result that 2/3rds of the change in \( z \) is caused by the change in \( x \).

Table 1: Components necessary to calculate the link score magnitude for the links \( x \rightarrow z \) and \( y \rightarrow z \) based on the equation \( z = x + y \).

| Variable | Time 1 | Time 2 | \( \Delta z \) | \( \Delta_x z \) | \( \left| \Delta_x z \right| \) |
|----------|--------|--------|----------------|----------------|-----------------|
| \( x \)  | 5      | 7      | 2              | 2/3            | 2/3             |
| \( y \)  | 4      | 5      | 1              | 1/3            | 1/3             |
| \( z \)  | 9      | 12     | 3              |                |                 |

Table 2 demonstrates why the absolute value function must be used to calculate the link score magnitude by showing how the sign of the link score magnitude term can yield incorrect results about the polarity of a link using the equation \( z = (w + x)/y \). The absolute value is part of the equation for the link score magnitude because the change in \( z \) may be positive or negative for reasons unrelated to the change in the input variable under study, in this case, \( x \). Essentially the direction of the change in \( z \) which influences the sign of the link score magnitude could be due to other inputs. In Table 2, incorrect polarities would be result from using the non absolute value link score magnitude because the change in \( y \) causes \( \Delta z \) to be negative which impacts the sign of the link score magnitude for all 3 links.

Table 2: Demonstration of wrong polarity when calculating the link score magnitude for the links \( w \rightarrow z \), \( x \rightarrow z \), and \( y \rightarrow z \) based on the equation \( z = (w + x)/y \).

| Variable | Time 1 | Time 2 | \( \Delta z \) | \( \Delta_x z \) | \( \frac{\Delta_x z}{\Delta z} \) |
|----------|--------|--------|----------------|----------------|-----------------|
| \( w \)  | 7      | 10     | 1              | -5             |                 |
| \( x \)  | 2      | 4      | 0.67           | -3.33          |                 |
| \( y \)  | 3      | 5      | -1.2           | 6              |                 |
| \( z \)  | 3      | 2.8    | -0.2           |                |                 |

Because of the erroneous results demonstrated in Table 2, the sign of the link score magnitude is ignored via the usage of the absolute value function. To measure polarity and finish the link score calculation we must multiply the link score magnitude by Richardson’s polarity \( \text{sign} \left( \frac{\Delta_x z}{\Delta x} \right) \). In Table 3 using the same equation and parameterization as Table 2 we apply our restatement of Richardson’s polarity formulation to get the correct polarities. To calculate the polarity for the link from \( x \rightarrow z \) start by following the procedure to calculate \( \Delta_x z \) from Table 1 and Table 2 above. Following that procedure yields a \( \Delta_x z \) of 2/3. Next determine the change in \( x \) which in this case is 2. Finally take the sign of \( \Delta_x z \) (2/3) divided by \( \Delta x \) (2) which is +1, and this is the correct polarity for the link \( x \rightarrow z \).
Table 3: Demonstration of correct polarity when calculating the link score magnitude for the links $w \rightarrow z$, $x \rightarrow z$, and $y \rightarrow z$ based on the equation $\Delta z / \Delta x = (w + x) / y$.

| Variable | Time 1 | Time 2 | $\Delta x$ | $\Delta z$ | $\text{sign} (\Delta z / \Delta x)$ |
|----------|--------|--------|------------|------------|-------------------------------------|
| $w$      | 7      | 10     | 3          | 1          | +1                                  |
| $x$      | 2      | 4      | 2          | 0.67       | +1                                  |
| $y$      | 3      | 5      | 2          | -1.2       | -1                                  |
| $z$      | 3      | 2.8    |            |            |                                     |

**Defining loop scores**

In order to establish a common baseline for comparing the relative strength of feedback loops, we need to identify which feedback loops to include in the comparison. To do that we introduce a refinement to Oliva’s SILS which we call the coupled SILS (CSILS). The CSILS is a collection of feedback loops taken from the SILS where every stock in the set has a path to and from every other stock in the set. Often there is only a single CSILS for an entire model and the SILS does not need to be further partitioned (well-connected models). But, as shown in the inventory workforce case below, that is not always true, and in that specific case (Figure 6) Expected Demand affects Inventory and Workforce, but is not itself affected by Inventory or Workforce. CSILS are necessary to make sure that we compare loops which affect stocks where the determinants of behavior for those stocks are shared. This is what allows the loop score to describe the percentage contribution of a feedback loop across multiple stocks.

The loop score is a normalized measure taking on a value between -1 and 1 computed using loop power as described below. It reports the polarity and instantaneous percentage contribution of a feedback loop to the change in behavior of all stocks in the CSILS it is a member of. By comparing loop scores we can determine which loops are dominant in the CSILS under study.

Loop power is the product of all of the link scores in the loop. Note that this multiplies both the magnitude and the sign of the different link scores, with an odd number of negative links yielding a negative loop. The product is used following the chain rule and this also accurately represents the effects of a dead link in an otherwise ‘active’ loop. This has the consequence of assigning any loop with a dead link a loop power (and consequently a loop score) of 0. The magnitude of loop power represents the force that a loop is exerting to change stock behavior across all stocks in its CSILS.

The loop score is computed by dividing the loop power by the sum of the absolute value of all loop powers in the CLIS. The sign of a loop score represents the polarity of the feedback loop.

This normalization is critical to maintaining scores that are easy to work with. Because of the definitions of link scores, loop power values can become very large as an equilibrium is approached. This is shown below for the bass diffusion model. In that case, even though the power of the loops effectively approaches infinity, the transition from positive to negative loop...
dominance is smooth and clearly visible when using the loop score because it is normalized. The concept of the CSILS is important only for this normalization process and ensures that loop power values are comparable. An example of incomparable loop power values is shown below in the case of the inventory workforce model where the feedback loop B3 is not comparable with the others since it is not coupled with B1 or B2.

Both the loop score and loop power concepts are rigorously defined below in equations (3a and 3b) where \( L_x \) refers to the loop \( x \) being studied. \( LS(i_1 \rightarrow d_1) \) refers to the link score for the first link in the loop from independent variable \( i_1 \) to dependent variable \( d_1 \). Link scores are multiplied for all links in the loop from that first link \( i_1 \rightarrow d_1 \) to link \( i_n \rightarrow d_n \) where \( n \) is the number of links in the loop \( x \).

\[
Loop\ Power(L_x) = (LS(i_1 \rightarrow d_1) \cdot LS(i_2 \rightarrow d_2) \ldots \cdot LS(i_n \rightarrow d_n))
\]

(3a)

\[
Loop\ Score_{L_x} = \left(\frac{Loop\ Power(L_x)}{\sum_{y=0}^{n} |Loop\ Power(L_y)|}\right)
\]

(3b)

**Computational considerations**

We make our computations as time progresses in the model. The first computation can be made only after the model has been initialized and moved forward in time. In the results we present, we use the model’s \( dt \) or time step to determine how often to compute link and loop scores, this is most straightforward using the Euler integration method. Conceptually the computation could proceed at a longer or shorter sampling interval allowing it to work with a non-fixed time step integration methods such as Runge-Kutta.

The computational efficiency of this method has not been examined in depth and it especially has not been analyzed in the case of large models with hundreds of stocks and millions of feedback loops. For models of smaller size and complexity (between 2 and 20 stocks) we have found through experience that the largest computational burden is not the calculations necessary for the link and loop scores, but rather the calculations necessary to identify the full set of feedback loops and the creation of the SILS before it is further partitioned (if necessary) into the coupled feedback loop set.

An implication of the structure of the calculation method we present is that the equations in the model will be computed not just once as it is typically done to solve an SD model, but again for each variable used in the equation. This can multiply the number of computations by 2 to 10 (depending on equation complexity) which is similar to the requirements for linearization but does not require any matrix decomposition as, for example, EEA does. In short, the computation times are modest and quite similar to simply simulating the model. For reference the analysis of all of the models in the paper, including Forrester’s 10 stock market growth model takes less than 1 second including the time to parse the XMILE representation of the
model, find all the loops, partition the loop set, and calculate all loop dominance metrics presented.

Computation of the link and loop score metrics, nonetheless, does require that equations be computed multiple times per dt and this can’t be done in standard software. For this paper we have modified an open source and publicly available simulation engine sd.js (Powers, 2019) to simulate the model and do the link and loop score calculations. Pseudo code for this computation is shown below in Figure 1.

```javascript
for (let target in model.variables) {
    let value = target.currentValue;
    let previousValue = target.previousValue;

    if (target.isStock) {
        let sumOfFlows = 0;
        for (let source in target.sources) {
            if (target.isInflow(source))
                sumOfFlows += source.previousValue;
            else
                sumOfFlows -= source.previousValue;
        }
        for (let source in target.sources) {
            if (sumOfFlows == 0) {
                LINKSCORE[source,target] = 0;
            } else if (target.isInflow(source)) {
                LINKSCORE[source,target] = -ABS(source.previousValue / sumOfFlows);
            } else {
                LINKSCORE[source,target] = ABS(source.previousValue / sumOfFlows);
            }
        }
    } else if (value == previousValue) {
        for (let source in target.sources) {
            LINKSCORE[source,target] = 0;
        }
    } else {
        for (let source in target.sources) {
            let tRespectSource = <calc. target, use current source, prev. of rest>;
            let deltaTRespectS = tRespectSource - previousValue;
            let deltaSource = source.currentValue - source.previousValue;
            let deltaT = value - previousValue;
            let sign = 1;

            if (deltaSource != 0 && deltaTRespectS!= 0) {
                sign = SIGN(deltaTRespectToS / deltaSource);
            }

            LINKSCORE[source,target] = ABS(deltaTRespectS / deltaT) * sign;
        }
    }
}
```

*Figure 1: Pseudo code for calculating all link scores in a model after calculating a dt of the model*
Application of the LTM method to the bass diffusion model

We’ve used a variant of the Bass diffusion model (Bass, 1969) pictured in Figure 2 below to demonstrate the ability of the LTM method to reproduce the standard explanation for the behavior of this model. Richardson (1995), using his method for determining dominant polarity says the following about how the Bass diffusion model, which is an example of a logistic equation, works:

*In the logistic equation, a shift in loop dominance occurs when the level reaches half its maximum value, the point of inflection in the logistic curve.*

This version of the Bass diffusion model runs from Time 0 to Time 15 with the inflection point reached between time 9.5625 and 9.625. It contains two loops, one balancing and one reinforcing.

![Diagram of Bass diffusion model](image)

*Figure 2: The stock and flow structure of the bass diffusion model analyzed*

- **Balancing (B1)**
  - probability of contact with potentials
  - potentials contacts with adopters
  - adoption from word of mouth
  - adopting
  - potential adopters

- **Reinforcing (R1)**
  - adopter contacts
  - potentials contacts with adopters
  - adoptations from word of mouth
  - adopting
  - adopters
Table 4 and Figure 3 demonstrate that LTM reproduces the same standard explanation for behavior as Richardson reports. In Table 4 the calculation of the loop power of B1 at specific points in time is demonstrated and compared to the loop power of R1 which shows that the two loops shift in dominance during the inflection point in behavior between time 9.5625 and 9.625. In addition, Table 4 confirms the proper polarity is assigned to each loop and link. Figure 2 supports the standard explanation of the model’s behavior by showing the loop score magnitude for both loops passes through .5, the threshold for dominance, during the inflection point.

### Table 4: Loop power in the Bass diffusion model calculated to 4 significant digits

| Link                                      | $T_1$ | $T_{9.5}$ | $T_{9.5625}$ | $T_{9.625}$ | $T_{15}$ |
|-------------------------------------------|-------|-----------|--------------|-------------|----------|
| Probability of contact with potentials → potentials contacts with adopters | 0.000 | 9.958     | 9358         | 10.91       | 1.000    |
| Potentials contacts with adopters → adoption from word of mouth               | 1.000 | 1.000     | 1.000        | 1.000       | 1.000    |
| Adoption from word of mouth → adopting                                            | 1.000 | 1.000     | 1.000        | 1.000       | 1.000    |
| Adopting → potential adopters                                                      | -1.000 | -1.000 | -1.000       | -1.000      | -1.000   |
| Potential adopters → probability of contact with potentials                      | 1.000 | 1.000     | 1.000        | 1.000       | 1.000    |
| **B1 Loop Power**                                                                  | 0.000 | -9.958    | -9358        | -10.91      | -1.000   |
| **R1 Loop Power**                                                                  | 1.000 | 11.46     | 9806         | 10.41       | 0.000    |

Table 4 identifies which links are most critical for explaining why feedback loop power changes. The only link score which shows a change in strength overtime in the loop B1 is the link ‘Probability of contact with potentials → potentials contacts with adopters’. Because this is the only link which has a changing score it is the key link in the loop B1 and is responsible for the changes in B1’s loop power as well as B1’s shift in loop dominance. Figure 1 shows that this link is located at the point of the non-linearity, the junction between the reinforcing and balancing feedback loops supporting the notion that this link is most important to the behavior of the loop B1, because it is only at the point where feedback loops meet that feedback loops are able to interact.
The loop power values in Figure 4 for the Bass diffusion model demonstrates why loop scores are normalized values and how loop power is used to gauge the overall effort the loops in a model are expending to change behavior. In this case when the flow values pass through 0, Figure 3 shows that the absolute value of loop power of both loops approaches infinity. The drastic change in scale makes Figure 3 not effective for quickly and accurately determining the dominant loops in the system, but it does demonstrate the absolute amount of the efforts to change the stocks by each loop at each point in time. When the loop power values for counteracting loops in a model are high, both loops are working hard to change behavior in opposite directions and therefore a small change in the stocks is observed.
The LTM analysis of the bass diffusion model has replicated the standard explanation for behavior in the model as performed by Richardson. In addition, the LTM analysis has identified the specific links in the loops which are most critical for explaining the shifts in feedback loop dominance and has identified the overall effort the loops in the model are expending to change behavior.

**Understanding the yeast alcohol model**
The yeast alcohol model is analyzed to demonstrate the efficacy of the LTM method. This analysis reinforces the notion that LTM is able to yield the same insights into behavior as previous analyses of this model using Ford’s behavioral approach, PPM and EEA (Saleh, 2002; Güneralp, 2006; Phaff et al., 2006; Mojtahedzadeh, 2008; Hayward and Boswell, 2014).

![Figure 5: Yeast alcohol model](image)

Figure 5 shows the structure of the model as analyzed which was done so using a DT of .5. The model structure; \( B = C \times (1.1 - 0.1 \times A) / b_1 \), \( D = C \times \exp(A - 11) / d_1 \), \( sAdt = p \times C \), is initialized as such; \( A = 0, B = 1, b_1 = 16, d_1 = 30, \) and \( p = 0.01 \). It contains 4 loops, all in a single feedback loop subset. Loop R, represents the birth\(^1\) of the cells C, characterized by the fertility, \( b_1 \). Loop B1 represents the natural death of the cells. The main link in Loop B2 represents the slowing of

\(^1\) Notice that there is a flaw in the standard formulation of B in this model causing B to take negative values and the polarity of R to change so that it acts as an additional “deaths loop” under conditions of high levels of alcohol A. This flaw has not been corrected in order to maintain the consistency of the model across analyses in the literature.
the birth of cells due to the presence of alcohol. The main link in Loop B3 represents the increasing death of cells due to the presence of alcohol. This model produces the overshoot and collapse behavior seen in Figure 5 which matches exactly the behavior generated by Phaff et al. (2006) and Mojtahedzadeh (2008), and is very similar to the results that and Hayward and Boswell (2014) generated using their slightly altered structure of the model.

Table 5: Dominant loops in yeast alcohol model.

| Time range | Phase 1: 0-51.5 | Phase 2: 52-66 | Phase 3: 66.5-75 | Phase 4: 75.5-100 |
|------------|-----------------|----------------|-----------------|------------------|
| Dominant loop | R               | B2             | B3              | B1               |

Table 6 identifies the dominant loops for each phase of the model’s behavior. Comparing these results with Ford’s (1999) behavioral approach as applied by Phaff et al. (2006), LTM identifies the same exact 4 phases and agrees with the behavioral analysis in principal, LTM’s only disagreement is that phase 3 is dominated by B3 alone, not B2 and B3. This same difference is

2 Hayward and Boswell (2014) use the same parameterization of the yeast alcohol model as us and the others, but appear to have used a Stella version of this model where uniflows were used for B and D. This subtle change to structure corrects the formulation flaw in the births loop (referenced in footnote 1) and causes their model results and loop dominance analysis to differ slightly from the other analyses and our own.

3 At time 74 no single feedback loop is dominant because this is the point where R is at its strongest as a balancing feedback loop (because of the model formulation error in footnote 1). After time 70 when the birth rate is negative R is acting in a similar fashion as B1. At Time 74 summing the strength of R & B1 yields a loop score which is stronger than B3, but still not over 50%, B3 is the single strongest feedback loop at that exact moment and we therefore consider it alone to be dominant across phase 3.
raised by the PPM approach of Mojtahedzadeh (2008) and Hayward and Boswell (2014) where PPM and the loop impact method identify that Phase 3 is dominated by B3 rather than B2 and B3 together as Phaff et al.’s implementation of Ford’s behavioral approach analysis would suggest. This shows that the LTM analysis of this model matches Ford’s behavioral approach with the noted discrepancy and matches exactly the PPM analysis done by Mojtahedzadeh. When we compare Table 6 to Hayward and Boswell’s (2014) PPM based loop impact method, we agree in principal with their results, but the slight change in the structure of their model prevents a true match.

Our results in Figure 6 and Table 6 match the EEA analysis of this model performed by Phaff et al. (2006). The EEA analysis done by Phaff et al (2006), shows that the behavior of Phase 1 is dominated by R with B2 restraining the growth of C. In phase 2 EEA shows that B2 is now dominant, but R is still a significant factor in explaining C which matches LTM and can be seen in Figure 6 because B2 has a loop score less then -0.5 and R is the only other active loop until time \sim 60 where B3 starts becoming active in preparation for phase 3. In phase 3, EEA points to B1 and B3 together as describing the behavior of C which is true according to the LTM analysis, with the caveat the LTM analysis finds that B3 is solely dominant throughout that time period. EEA then reports that during phase 4 B1 is dominant over B3 which again matches the LTM analysis and can be seen in Figure 6 as B1 starts growing quickly at the end of phase 3 reaching a loop score of nearly -1 shortly after the start of phase 4.

The LTM analysis of the yeast alcohol model demonstrates a shared common understanding of the model by EEA, and PPM, both automated loop dominance analysis techniques, in addition to Ford’s behavior analysis technique demonstrating that LTM is capable of yielding the same level of insight into this model as these other techniques.

**Using LTM to understand oscillations**

Figure 8 shows that LTM identifies oscillation as the result of a single feedback loop in Gonclaves (2006) adaptation of Mass and Senge’s (1975) two state inventory workforce model pictured in Figure 7. This result is the major defining characteristic which distinguishes the analyses of models by LTM and EEA from those done by PPM based methods. We compare and contrast the results of the LTM analysis with the EEA analysis of the same model by Gonclaves (2009), and by two PPM based analyses of the original Mass and Senge model by Mojtahedzadeh (2008) and Hayward and Roach (2018). The only difference between Gonclaves (2009) model and Mass and Senge’s (1975) model is the addition of feedback loop B3 in Figure 7 which does not significantly affect the analysis of the cause of the oscillation, only the exact shape of the oscillatory mode of behavior. In addition, the LTM analysis of this model is used to demonstrate the impact of parameters on loop dominance patterns.
Our implementation of Gonclaves (2009) model runs from Time 0 to Time 60. The model has only three balancing feedback loops that appear in two different BO-SILS which demonstrate the inability of B1 or B2 to influence B3. The method for determining the two CSILS is specified in the section ‘defining loop scores’. The graphical function inside of the ‘demand’ variable acts like a step function, triggering a single increase in demand between times 1 and 2 which sets off a dampened oscillation in both workers and inventory.

- **Set 1**
  - **Major Balancing (B1)**
    - Inventory
    - inventory gap
    - desired change in inventory
    - desired production
    - desired workers
    - workers gap
    - hiring or firing
    - Workers
    - producing
  - **Minor Balancing (B2)**
    - Workers
    - workers gap
    - hiring or firing
The two loops in this model that contain the stocks with the oscillatory behavior are B1 and B2 of Set 1. As shown in Figure 8, in all three parameterizations of the model the dominant loop describing the large majority of the change in the behavior of the worker and inventory stocks, and therefore the oscillations in the model is the major balancing loop B1. Figure 8 shows that the strength of B2 which is driven by the adjustment to the time to hire or fire parameter affects the shape of the dampened oscillation even though B1 is responsible for the oscillation itself. The longer B2 is active the more pronounced the oscillations are, and this tells us that by increasing the time to hire or fire, we increase the strength of B2 (relative to B1). While it is true that time to hire or fire is strongly tied to the strength of B2, it also directly affects the strength of B1 independently of its effects on B2. Figure 8 shows the total relative effects of this parameter change on the strengths of B1 and B2. The sum total of the effects of time to hire or fire on B1 and B2 cause the changes in relative strength which causes the dampening of the oscillations to slow which causes the oscillations to become more pronounced and last longer.

The LTM conclusions about the impacts of time to hire or fire on the oscillation matches both the EEA and PPM derived conclusions, but differs from the PPM based analyses done by both Mojtahedzadeh (2008) and Hayward and Roach (2018) in its explanation for how those conclusions were reached. The PPM based methods show the oscillations as arising from a process dominated by both B1 and B2 in a cyclical nature. The LTM interpretation matches closer to Gonclaves (2009) EEA analysis which shows that the oscillatory mode of behavior arises primarily from the loop gains associated with the impacts from the major loop B1 and the damping effect is a function of B2. This line of reasoning is further supported by Kampmann and Oliva (2008) who point out that Mojtahedzadeh’s PPM method is destined to mis-handle oscillation in sinusoidal linear systems like the inventory workforce model, by producing explanations which show shifting feedback loop dominance because of sign changes in the PPM itself even though the relative strength of the loops in the model remains constant. Also of note in Figure 8 is the time period before the shock in demand. During that period the model is in equilibrium, unchanging, and therefore LTM cannot inform or analyze the model because all link scores are 0.

The LTM analysis of the inventory workforce model has demonstrated the efficacy of the LTM method for analyzing oscillatory systems. Its demonstrated that the LTM method produces causal explanations for oscillations more akin to EEA analyses then PPM based analyses, and produces conclusions about the effects of the time to hire or fire parameter that matches both.
Figure 8: Results of LTM analysis of the Inventory Workforce model showing the effect of time to hire or fire on loop dominance and Workers.

Discussion and conclusions
As demonstrated, the LTM method provides an easy way to understand and identify, through computation, which feedback loops in a model dominate or in other words describe more of the behavior at each point in time. Dominance is CSILS wide, based on the effect on all variables, and is typically driven by the stocks that are changing proportionally most quickly, and the feedback acting in support of that movement. As we have seen in the examples, this measure of dominance correlates well with our structure-based understanding of relatively simple systems. The LTM method has several considerable advantages outlined here:

Most importantly, this method is generally applicable to all models without any manipulation or modification and the format of the results of the analysis are simple, easily interpretable graphs of behavior over time. LTM makes use of the existing skillsets of all modelers and most model
consumers and is thus easily accessible. As for its general applicability, at present the current implementation of the LTM method works with any non-arrayed XMILE compliant model containing the most common built in functions and graphical nonlinear relationships. Since the method uses only values calculated by the model as it runs, the structure of the model never needs to change to accommodate LTM analyses. Finally, it’s a major benefit of LTM that analyses are conducted over time in lock step with behavior. This creates data and data visualization opportunities which match the current structure of the information taken in by those who already employ SD modeling. Because the LTM results are calculated and reported over time it takes no additional skill or training to be able to read and interpret the graphs of loop score, loop power and link score.

The second key advantage to the LTM method is its relative simplicity. The method as currently developed does not use complex mathematical constructs which are not already in use by the majority of practitioners. From a mathematical perspective the concept of the $\Delta_{x}z$ is the most difficult part of the method because of its unfamiliar terminology, and not necessarily because of any inherent complexity in the idea itself. The advantage of a simpler method is that it can be understood by all practitioners so that when it comes time to apply the method, practitioners can know ‘what it is doing’ due to the transparency of the method.

The third key advantage is that this method is relatively easy to implement in existing simulation engines without requiring modification to existing structures within those engines. We base this conclusion upon our own experiences implementing simulation engines in the past including the engines behind Stella and Vensim, and the level of effort it took to modify Powers’ sd.js engine. This means its uptake should be relatively painless by software vendors in the field if they so choose.

In addition to the key advantages listed above, the LTM method allows for the development of new and exciting visualization tools including animated stock-and-flow diagrams where the links and flows change color and size due to changes in polarity or link strength, in response to Sterman in Business Dynamics (2000). Going even further, the LTM method allows for the possibility of automated CLD generation and animation. Because the LTM method is able to say on a link-by-link basis which are the key (dynamic) links in the model, it is possible, using the method, to automatically generate a CLD collapsing all of the ‘unimportant’ static links with scores of 0, +1.0, or -1.0 into links which are conveying a change, which exist at the junction points of the loops. This will allow for the automated generation of structurally correct, minimal CLDs that accurately portray the structural components that predominantly produce the dynamics of the model, and are laid out by the computer automatically according to best practices.

While using only computed variable values is a strength of LTM, it also means that only realized, and never latent, model behavior can be analyzed. Thus, unlike EEA techniques, LTM on its own is unable to provide a general model level of understanding. Some of this understanding, including behavioral sensitivity to parametric changes, can be gained through a combination of sensitivity and LTM analysis as suggested below.
Additionally, the LTM method is not able to determine loop dominance without a change in the model state. As models approach equilibrium, we can see the loop scores balance one another even as they become unbounded, but when a model is in equilibrium all loop scores are 0. Therefore, models in equilibrium cannot be analyzed using LTM. An example of this is a simple “bathtub” population model where the birth fraction equals the death fraction. The limitations of the link score causes the loop power for both loops to be 0 because there is no change across a timestep, dt. An unsatisfactory solution to this problem from a purely methodological perspective is to start introducing minute changes in these situations in order to measure those changes impacts on loop dominance, but doing that would have major ramifications on the utility of the method for discrete and discontinuous models where information is likely encoded as specific logically meaningful integer values. Currently, models in equilibrium are much better analyzed using EEA methods like those suggested by Oliva (2016). An alternative approach would be for the model author to offset the model state from its equilibrium using a STEP function or other modeling construct for making constants vary due to exogenous forces.

An additional strength and weakness of the LTM method is that it focuses exclusively on endogenously generated behavior. Such a focus is a hallmark of System Dynamics, but is problematic for models where behavior is driven through external forcing functions that dominate the effects of feedback in the model. Loop dominance, in this case, may have little to do with behavior generated. Models of this sort are currently much better analyzed using the loop impact method of Hayward and Boswell (2014).

There are a variety of interesting extensions to LTM that combine it with other analysis techniques. The most obvious one is to combine it with sensitivity analysis so that the realized behavior sets encompass the potential behavior sets. For example, using extreme condition testing could, combined with LTM, be used to show that the model is producing the right results for the right reasons. LTM could also be combined with optimization, for example using optimizers to maximize or minimize loop scores. This would allow practitioners to maximize the impact of favorable loops while minimizing the impact of unfavorable loops in order to automatically generate better, more robust policy recommendations. Another area of study would include loop scores in the outputs of Monte-Carlo sensitivity analyses which would allow us to measure the robustness of loop dominance to policy or parameter changes. Monte Carlo analysis could also be used to measure the sensitivity of loop power to changes in parameter values.

Finally, it is necessary to test and analyze larger and more varied models if we are to increase our confidence in the general utility of the LTM method. We’re hopeful that the techniques laid out in this paper, will offer a significant utility and enhance the analysis and understanding of a wide set of SD models by all level of SD users.

References
Bass, F. M. (1969). A new product growth for model consumer durables. Management science 15(5): 215-227.
Checkland P, Scholes J. 1990. Soft Systems Methodology in Action. Wiley: Chichester.
Eberlein RL. 1984. Simplifying dynamic models by retaining selected behavior modes. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Davidsen P. 1991. The Structure-Behavior Graph. The System Dynamics Group, MIT, Cambridge, MA.

Ford D. 1999. A behavioral approach to feedback loop dominance analysis. System Dynamics Review 15(1): 3–36.

Forrester N. 1982. A Dynamic Synthesis of Basic Macroeconomic Theory: Implications for Stabilization Policy Analysis. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Graham AK. 1977. Principles of the Relationship Between Structure and Behavior of Dynamic Systems. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Goncalves P. 2009. Behavior modes, pathways and overall trajectories: eigenvector and eigenvalue analysis of dynamic systems. System Dynamics Review 25(1): 35–62.

Güneralp B. 2006. Towards coherent loop dominance analysis: progress in eigenvalue elasticity analysis. System Dynamics Review 22(3): 263-289.

Hayward J, Boswell GP. 2014. Model behaviour and the concept of loop impact: a practical method. System Dynamics Review 30(1): 29–57.

Hayward, John, and Paul A. Roach. 2017. Newton's laws as an interpretive framework in system dynamics. System Dynamics Review 33(3-4): 183-218.

Kampmann CE. 2012. Feedback loop gains and system behaviour (1996). System Dynamics Review 28(4): 370–395.

Kampmann CE, Oliva R. 2006: Loop eigenvalue elasticity analysis: Three case studies. System Dynamics Review. 22(2): 141-162.

Kampmann CE, Oliva R. 2009. Analytical methods for structural dominance analysis in system dynamics. In Meyers RA (ed.) Encyclopedia of Complexity and Systems Science Springer, New York NY, 8948–8967.

Mass, N. J., & Senge, P. M. (1975). Understanding oscillations in simple systems. System Dynamics Group Working Paper D-2045-2, MIT.
Mojtahedzadeh MT. 1996. Structural Analysis of the URBAN1 model. University at Albany, SUNY, NY 12222, USA.
Mojtahedzadeh M, Andersen D, and Richardson GP. 2004. Using Digest to implement the pathway participation method for detecting influential system structure. *System Dynamics Review* 20(1): 1-20.

Mojtahedzadeh M. 2008. Do parallel lines meet? How can pathway participation metrics and eigenvalue analysis produce similar results?. *System Dynamics Review*. 24(4): 451-478.

Moxnes, Erling, and Pål I. Davidsen. 2016. Intuitive understanding of steady-state and transient behaviors. *System Dynamics Review* 32(2): 128-153.

Oliva, R. (2004). Model structure analysis through graph theory: partition heuristics and feedback structure decomposition. *System Dynamics Review*, 20(4): 313-336.

Oliva R. 2016. Structural dominance analysis of large and stochastic models. *System Dynamics Review* 32(1): 26-51.

Phaff HWG, Slinger JH, Güneralp B, and van Daalen CE. 2006. Investigating model behavioural analysis: A critical examination of two methods. In *Proceedings of the 24th International Conference of the System Dynamics Society*. Nijmegen, The Netherlands, System Dynamics Society.

Powers, R. (2019). SD.js in-browser system dynamics model simulation and display. GitHub repository: [https://github.com/bpowers/sd.js](https://github.com/bpowers/sd.js)

Richardson GP. 1995. Loop polarity, loop dominance, and the concept of dominant polarity. *System Dynamics Review* 11(1): 67–88.

Richardson GP. 1996. Problems for the future of system dynamics. *System Dynamics Review* 12(2): 141-157.

Sterman JD. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill, Boston.

Saleh MM. 2002. The Characterization of Model Behavior and its Causal Foundation. PhD Thesis, University of Bergen, Bergen, Norway.

Saleh M, Oliva R, Kampmann CE, Davidsen P. 2010. A comprehensive analytical approach for policy analysis of system dynamics models. *European Journal of Operational Research* 203(3): 673-683.