

Uncertain Hierarchical Modeling

Sven Khatri Raff D'Andrea John Doyle
Control and Dynamical Systems 116-81
California Institute of Technology
skhatri@hot.caltech.edu

Abstract

For modeling complex systems, it is natural to reduce the system into subsystems and model each subsystem. The approach taken in this paper is that it is desired that a model should be consistent with the modeling methodology. Further it is important to explicitly represent the inaccuracies of the model as part of the model.

Within this paper, uncertain hierarchical modeling is further motivated. A hierarchy, interconnection structure, and a fundamental component data type are proposed and the choices motivated. The framework is proposed with the intention of being implemented on a computer and having a family of models of different resolution representing a system.

Keywords: Hierarchical modeling, uncertainty, implicit systems

1 Introduction

A modeling framework involves choices. This paper presents and motivates the choices made in this uncertain hierarchical modeling framework. The motivation for choosing hierarchical modeling is that not all components of a system are equally significant and this should be reflected in the model. In the case of a car, the ashtray is a less significant component than the engine. If a hierarchical model is not used then all the system equations are written at the same level, and it may be difficult to immediately distinguish the dominant dynamics of the system from the trivial dynamics. The hierarchy hopefully provides a quick and efficient method for identifying the significance of dynamics and doing model reduction as desired.

The second choice made is doing uncertain modeling. For a real system, finding an exact model is impossible and characterizing the inexactness is critical for making guarantees on system performance. In the case of a resistor, there is uncertainty on the exact value of the resistance and the parasitics. When a model is reduced, the reduced dynamics must be covered with uncertainty. This may be useful when cruder models with uncertainty are sufficient for a particular application which may lead to reduced computation.

The next choice made is a tree structured hierarchy.

The motivation for a tree structure is that a complicated system is naturally decomposed into an interconnection of simpler, more tractable subsystems and each subsystem can be similarly reduced. This self similarity leads to a tree structure. For example, modeling all the facets of a car is quite a task, but a more natural approach is to break up the car into more tractable components and model them individually. So rather than modeling the entire car, a car is an interconnection of an engine, transmission, exhaust system, cooling system, suspension, etc and each of the components is modeled separately.

A benefit of the tree structure is the connection with the object oriented philosophy. The tree defines how the components interact with each other. Which simplifies future modifications, because if a system is modified only the modified components needs to be remodeled, and the other component models will remain intact. So in the car example, if the engine is replaced with a different model, the entire system doesn't have to be remodeled. The old engine model is replaced by the model of the new engine.

Another choice that is made is that the model will be constructed, implemented and used on a computer. As a result, the data structures should be convenient and tractable for computer implementation as opposed to writing them out by hand. The reason for this choice is that as more complicated systems have more and more detailed models and are to be analyzed or simulated it is intractable to do them any other way.

The proposed framework is defined by a hierarchical tree structure of the components, an interconnection structure of the components, and a fundamental data type for a component. Throughout this paper, an inductor is used to demonstrate the features of the framework. An inductor is a simple example, but is necessary to make the presentation tractable.

2 Background

2.1 Linear Fractional Transformations

A linear fractional transformation is shown in Figure 1 for the map $y = (\Delta \star M)u$ where $\Delta \star M = D + C\Delta(I - A\Delta)^{-1}B$. In general, Δ represents uncertainty and dynamic elements, and $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is a realization of the map $\Delta \star M$. The LFT framework results

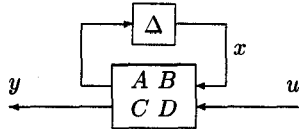


Figure 1: Linear Fractional Transformation

in a convenient method for adding various types of uncertainty to systems [11].

If $\Delta = f$ operator then the system shown in Figure 1 is just the standard state space representation. LFT systems are a natural generalization of state space representations. By allowing Δ to represent more general systems operators, LFT systems provide a convenient framework to add various types of uncertainty operators, like Non-linear, LTV, LTI, LPV, etc [6] [10] [3], in which essentially all the major state space results can be generalized [1].

2.2 Implicit LFT Systems

An implicit LFT system is described by $0 = (\Delta \star M)w$ as shown in Figure 2, where w contains all the system variables ie. there is no distinction between inputs and outputs. Implicit LFT systems are a generalization of the behavioral framework proposed by Willems [9].

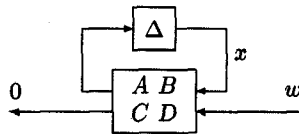


Figure 2: Implicit LFT system

Any LFT system can be converted into an implicit LFT system. In first principles modeling, like $F = Ma$, neither F or a is assumed to be an input or an output. Once either F or a is defined, both are defined. This is natural within the implicit LFT form but doesn't fit and makes interconnections difficult within the LFT form.

For tree structured hierarchical models, at the interconnections there isn't a notion of signal flow. The interconnection variables become internal variables to the system rather than inputs or outputs. So it is more natural to not make the distinction between inputs and outputs in modeling.

2.2.1 Interconnection of Implicit LFT Systems

Within the implicit LFT framework, the interconnection of systems is simple. The implicit description of a system describes the equations a system must satisfy ($0 = (\Delta_i \star M_i)w_i$ where $M_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$). So if two systems are connected then they still satisfy the same equations, and in addition the interconnection must be defined ($T_1w_1 + T_2w_2 = 0$) which defines the intersection

of behaviors [9]. So the implicit LFT model of the interconnected system is given by $0 = (\Delta \star M)w$ where:

$$M = \begin{bmatrix} A_1 & 0 & B_1 & 0 \\ 0 & A_2 & 0 & B_2 \\ C_1 & 0 & D_1 & 0 \\ 0 & C_2 & 0 & D_2 \\ 0 & 0 & T_1 & T_2 \end{bmatrix} \quad (1)$$

$$\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} \quad (2)$$

Implicit representations can be interconnected, manipulated and reduced without committing to a particular input-output form, which is only relevant to certain applications, and which can be easily derived a posteriori if necessary as shown in D'Andrea and Paganini [2].

2.2.2 Integral Quadratic Constraints

The implicit LFT framework also allows our model to include integral quadratic constraints (IQCs)[5]. IQCs are inequalities involving a quadratic form in signal space:

$$\langle \Pi w, w \rangle = \int_{-\infty}^{\infty} w(\omega)^* \Pi w(\omega) d\omega \leq 0 \quad (3)$$

where Π^* is an LTI operator. IQCs can be used to define sets in which signals must exist, like defining sets of allowable noise. Given $\Pi(e^{j\omega}) = \Pi(e^{j\omega})^* \in L_\infty \Rightarrow \exists k > 0 \ni kI + \Pi > 0$. By doing a spectral factorization [11] of $kI + \Pi$, we find a $Q \ni \Pi = kI - Q^*Q$. Defining $P = k^{1/2}I \Rightarrow \Pi = P^*P - Q^*Q$. Equation 3 becomes $\|Qw\|_2 \geq \|Pw\|_2$. This can be written as an uncertain implicit equation of the form $(P + \Delta_c Q)w = 0$ where Δ_c is an arbitrary contractive operator [7].

3 Component Modeling

Consider the ideal inductor shown in Figure 3 with the equation $d/dt(Li) = v$ with inductance L , where v is an

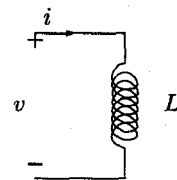


Figure 3: Inductor

input, i is an output, and ϕ is the flux in the inductor. The resulting LFT model (Figure 1) of the system is given by $u = v, x = \phi, y = i, \Delta = f$, and $\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1/L & 0 \end{bmatrix}$. As a result of choosing this model of an inductor, two problems arise. First, if the effects of uncertainty in L , nonlinearities and parasitics, are to be investigated, there is no convenient way to do this without starting over. Second, by assuming that v is an input and i is an output, our

model may be incompatible with other components with which it is interconnected as shown in Figure 4. Figure 4 represents an input/output interpretation of two inductors connected in series ($i_1 = i_2$). The connection

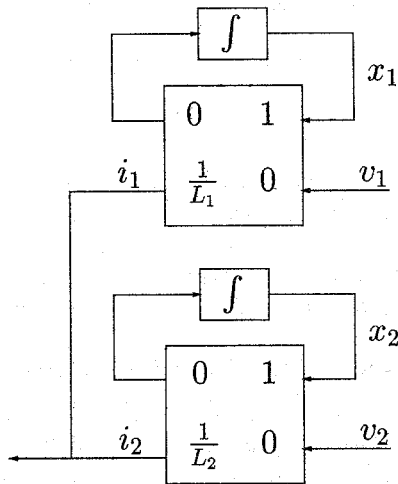


Figure 4: Series Interconnection of Inductors

of the two outputs is not properly defined within the input/output framework. These two problems are addressed in Sections 3.1 and 3.2.

3.1 Problem 1: Uncertainty

The solution to the first problem is easily addressed in the LFT framework. For the inductor example in Figure 3, the modeled equations should be replaced by $d\phi/dt = v$ and $\phi = L(I) = L_0(1 + \delta)i$. Where L_0 is a constant which represents the nominal value of the inductor and δ is an unknown operator. The model of the uncertain inductor is given by $y = i$, $x = [\phi, x_2]$, $u = v$,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/L_0 & -1/L_0 & 0 \\ 1/L_0 & -1/L_0 & 0 \end{bmatrix}, \text{ and } \Delta = \begin{bmatrix} \int & 0 \\ 0 & \delta \end{bmatrix}. x_2 \text{ is not}$$

a state in the conventional sense but the output δ [1]. δ could represent time variation in the inductance due to saturation, variations in the core geometry, magnetic links to other components, nonlinearities, etc.

3.2 Problem2: Interconnection Compatibility

LFTs provide a flexible modeling framework for incorporating uncertainty descriptions, but the input-output assumption is not desirable for the modeling of interconnected systems. It is not a priori known which variables should be treated as inputs and which should be treated as outputs [9].

To address the second problem, systems will be represented in an implicit LFT form where $0 = (\Delta * M)w$ as shown in Figure 2. There is no issue of compatibility

between implicit LFT models of components because no input-output partition has been made.

For the inductor example, the implicit model of the inductor is given by $d\phi/dt = v$, $L_0(1 + \delta)(i) = \phi$, $w = [v, i]$,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0 & L_0 & 0 & L_0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix}, \text{ and } \Delta = \begin{bmatrix} \frac{d}{dt} & 0 \\ 0 & \delta \end{bmatrix}.$$

3.3 Generality

For the fundamental components in this modeling system, the model must be general enough to describe any situation which may occur. The model should have some information about the component, so that each component isn't just an arbitrary operator. The information is contained in the nominal value of the component. No irreversible assumptions should be made. For example, the standard circuit equation for an inductor is $V = L \cdot di/dt$. If this were the fundamental model of an inductor, the assumption has been made that L is a constant. If L were blindly replaced by $L(t)$ the resulting model would be incorrect. The actual equations for an inductor are given in equations 4 and 5.

$$\phi = Li \quad (4)$$

$$kv = d\phi/dt \quad (5)$$

Once we have our model and are ready to do analysis, synthesis, or simulation any assumptions can be made about our uncertainty like Δ_L is a real parameter, a bounded operator, etc, but this is after the modeling process and a part of the identification process.

We want to make the weakest possible assumptions about our inductor (L, k), so that wide variety of uncertainty assumptions can be made at the analysis level. L and k are assumed to be a non-commuting indeterminates (NCIs), ie. it could be an arbitrary nonlinear, time-varying operator. For a real inductor, it's nominal value is of use in describing it's operation. For modeling, $L := L_0 + \Delta_L$ and $k := 1 + \Delta_v$, where Δ_L and Δ_v are NCIs and L_0 is a "place holder" for the nominal inductance. L_0 is used to describe the ideal model of an inductor. NCI's act as "place holders" for uncertainty descriptions. It is important to note that by setting $L := L_0 + \Delta_L$ we have not committed to anything. This can be undone by defining $\Delta_L := -L_0 + \Delta_{Lnew}$. The L_0 term is added because the nominal value is presumably useful in describing the operation of the system.

3.4 Model Modifications

Part of the modeling process is the addition on new components (like adding parasitics to the model of a inductor). It is desired that the current model can be refined to arrive at a new model rather than discarding

the model and starting over. As a result the model must have “place holders” for defining new interconnections. If these “place holders” do not exist then it is possible that at an interconnection an incorrect equation may result as shown in Figure 5. Figure 5a would lead to the con-

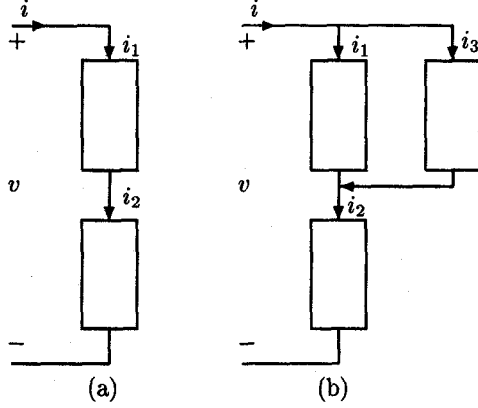


Figure 5: Incorrect and Irreversible interconnections

straint that $i_1 = i_2$ but if later an additional component is added as in Figure 5b then $i_1 = i_2 + i_3$. These two constraints lead to $i_3 = 0$ which may be incorrect.

In the circuit case, these “place holders” would correspond to error currents for each node and error voltages for each branch connecting nodes. These are needed to describe all the possible interconnection of circuits. When we do the analysis these error place holders must be resolved. They can be set to 0, ie. no additional current into a node, they can be considered external noise terms and constrained according to IQCs, or used to cover unmodeled dynamics.

4 Component Data Type

A component is chosen to be modeled by an implicit LFT system (Figure 2) with special structure. The implicit LFT structure is chosen because it provides a powerful framework for describing uncertainty plus it is natural for interconnecting systems. The system variables w are partitioned as:

$$w = [w_m \ w_n \ l_m \ l_c \ l_e] \quad (6)$$

Where w_m are the manifest variables of the component. They represent the variables that are connected to the parent component. For an inductor, $w_m = [v \ i]$. w_n are the nominal manifest variables, and l_m are the latent manifest variables. w_n are used to describe the nominal dynamics of the component. w_n and l_m are used to describe the interconnection of the nominal component with the child components. l_c are the latent crosstalk variables, and are the terms used to describe crosstalk connections (connection between unrelated components). l_e are the latent error variables, they are used to describe

interconnection errors and act as “place holders” for describing future interconnections. It is difficult to motivate this partition of the system variables, but this partition should be clear after the hierarchy and interconnection structure of the system is defined.

The special structure of $\Delta \star M$ is described by:

$$B = [0 \ b_1 \ 0 \ b_2 \ b_3] \quad (7)$$

$$C = \begin{bmatrix} c_1 \\ 0 \end{bmatrix} \quad (8)$$

$$D = \begin{bmatrix} 0 & d_1 & 0 & d_2 & 0 \\ d_m & d_n & d_i & d_c & d_e \end{bmatrix} \quad (9)$$

$$\Delta = \text{diag} \left(\frac{d}{dt} I, q_1 I, \dots, q_n I, \Delta_1, \dots, \Delta_n \right) \quad (10)$$

$A, B, C,$ and D are constant matrices. q_i are arbitrary constants. Δ_i are noncommuting indeterminates. Δ is used to describe the dynamics and uncertainty of the nominal component. Within Δ the standard dynamic element is $\frac{d}{dt}$ instead of the standard \int . The reason for this choice is that \int is not an operator because an initial condition must be specified.

The component model of a “real” inductor is shown in Figure 6. The implicit LFT model is given by (using

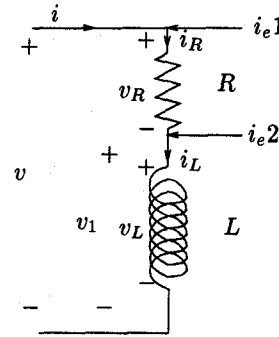


Figure 6: A Model of a “Real” Inductor

Figure 2, equations 7, 8, 9, and 10: $w_m = [v, i], w_n = [v_L, i_L], l_m = [v_R, i_R, v_1], l_c = [], l_e = [v_{e1}, v_{e2}, i_{e1}, i_{e2}, \phi_e]$ (v_{e1} corresponds to a voltage error of R , v_{e2} corresponds to a voltage error of L , and ϕ_e corresponds to the flux error in the inductor),

$$x = \begin{bmatrix} \frac{d\phi}{dt} \\ L_0(I + \Delta_L)i_L \\ \Delta_v v_L \\ \Delta_L i_L \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, b_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$c_1 = [-1 \ 0 \ 1 \ 0], d_1 = [1 \ 0],$$

$$d_m = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, d_n = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, d_l = \begin{bmatrix} -1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$d_e = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}, \text{ and } \Delta = \begin{bmatrix} \frac{d}{dt} & 0 & 0 & 0 & 0 \\ 0 & L_0 & 0 & 0 & 0 \\ 0 & 0 & \Delta_v & 0 & 0 \\ 0 & 0 & 0 & \Delta_L & 0 \end{bmatrix}$$

The terms that are not listed are empty ie. = \square .

5 Hierarchy of Components

The hierarchical structure of a system modeled using this framework is a tree (Figure 7). At each node of the tree there is a component of the form presented in section 4. Components can be ranked hierarchically if they are on the same branch of the tree. The component closer to the root is considered hierarchically above the component closer the leaf of a branch. Components not on a common branch are not related hierarchically. A particular branch may be more significant than another. For the inductor model in Figure 6, the hierarchy is shown

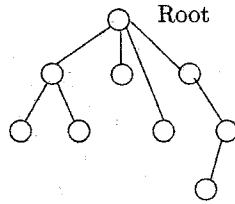


Figure 7: Hierarchical Structure

in Figure 8.

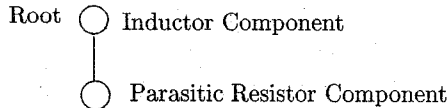


Figure 8: Hierarchical Structure of Inductor Model

The link that connects two nodes on the tree contains the information of how the latent variables of the higher component are connected to the manifest variables of the lower component. This is a hierarchical interconnection. The link defines T_1 and T_2 of (1) for the interconnection. Assuming that M_1 is the lower component and M_2 is the higher component, $T_1 = [I \ 0 \ 0 \ 0 \ 0]$ and $T_2 = [0 \ 0 \ t_2 \ 0 \ 0]$. For the inductor model, $t_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$.

6 Nonhierarchical Interconnections

Hierarchical interconnections are not the only interconnections. Any two nodes may be interconnected. The nonhierarchical interconnections are called crosstalk. Crosstalk is used to model secondary interactions between components like magnetic interconnection of disjoint inductors, gravitational interaction of masses, two disjoint flexible structures linked by air, etc. The structure of the model of the system is a web (Figure 9) rather than a tree, but the web has an underlying tree structure from the hierarchy as shown in Figure 7. The solid lines in

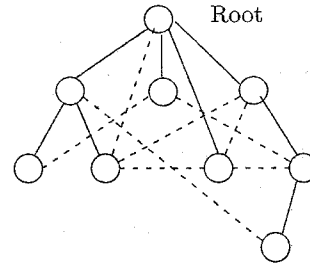


Figure 9: Web Structure of Model

Figure 9 denote hierarchical connections and the dashed ones the crosstalk connections.

A non-hierarchical interconnection link is actually a component describing the dynamics of the connection (dynamics of air linking two disjoint flexible structures). The link defines how the latent crosstalk variables of the two hierarchical components are connected to the crosstalk component. The link defines an interconnection of 3 components. So generalizing (1) we have T_1 , T_2 , and T_3 where $T_i = [0 \ 0 \ 0 \ t_i \ 0]$ for $i=1$ or 2 and $T_3 = [t_3 \ 0 \ 0 \ 0 \ 0]$.

7 Component Refinement

The process of component refinement is an improvement of the model of a component. A more accurate nominal model of the system (without uncertainty, noise, etc) can be parametrized. As a result, the uncertainty description of the system doesn't have to be as conservative.

Component refinement involves interconnecting a component to more "parasitic" subcomponents or describing crosstalk interaction between components, which models previously undescribed phenomenon. The proposed hierarchical modeling framework was designed with this option in mind. The idea being that a model can describe new phenomenon without having to start over. This is the purpose of the latent error variables l_e .

As a model is refined new latent manifest variables are created for components. In the circuit case these are

the new interconnection variables, the voltages associated with new nodes, the error terms for new nodes and branches, and refinements of old error terms.

In the process of refining the components of a model (arriving at a more detailed model) there is an implicit reduction of the uncertainty necessary to describe the system dynamics because the uncertainty is still represented by an NCI. As more phenomenon are modeled, the inaccuracy of the model is reduced.

8 Model Reduction

Assuming that we have a detailed model of a system with all the nominal values determined, NCIs characterized, and error variables resolved, it may be desirable to view the system at different levels of resolution. There is a tradeoff between the complexity and fidelity of a model. When the model is used for analysis, synthesis of a controller, or simulation less detailed models may be sufficient for the task at hand which would reduce the required computation. There is no reason to use highly detailed models when crude models are sufficient.

In this framework, model reduction can be done efficiently. The model reduction process involves paring the hierarchical tree. When a branch is cut the dynamics of the removed components must be covered by uncertainty. The removed branch would be replaced by IQCs.

9 Conclusions

In this paper, an uncertain hierarchical modeling framework, including a component data type, interconnection structure, and hierarchical structure, is proposed and motivated. The framework is appropriate for complex systems and computer implementation and provides for multiresolution models which can be changed as desired. In the analysis and simulation of large complex systems, this framework would be vital to attacking the problem in a coherent manner and providing a tractable approach.

10 Acknowledgements

This work was supported by AFOSR and the Computational Science Graduate Fellowship Program of the Office of Scientific Computing in the Department of Energy.

References

- [1] Carolyn Beck, Raffaello D'Andrea, Fernando Paganini, Wei-Min Lu, and John Doyle. A state space theory of uncertain systems. In *IFAC 1996*, 1996.
- [2] Raffaello D'Andrea and Fernando Paganini. Interconnection of uncertain behavioral systems for robust control. In *32nd IEEE CDC*, 1993.
- [3] Raffaello D'Andrea and Fernando Paganini. Why behave? In *32nd IEEE CDC*, 1993.
- [4] John Doyle, Fernando Paganini, and Carolyn Beck. Approximate behaviors. In *35th IEEE CDC*, 1996.
- [5] A. Megretski and S. Treil. S-procedure and power distribution inequalities: A new method in optimization and robustness of uncertain systems. *Mittag-Leffler Institute*, 1991.
- [6] A. Packard and J. C. Doyle. The complex structured singular value. *Automatica*, 29(1):71-109, 1993.
- [7] F. Paganini. *Sets and Constraints in the Analysis of Uncertain Systems*. PhD thesis, California Institute of Technology, 1995.
- [8] Fernando Paganini and John Doyle. Analysis of implicitly defined systems. In *33rd IEEE CDC*, 1994.
- [9] Jan C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 1991.
- [10] P. M. Young, M. P. Newlin, and J. C. Doyle. Let's get real. CDS Technical Memo CIT-CDS 92-001, California Institute of Technology, Pasadena, CA 91125, September 1992.
- [11] Kemin Zhou, Keith Glover, and John Doyle. *Robust and Optimal Control*. Prentice Hall, 1995.