Estimation of Signals in an Interconnection of LTI Systems and Unknown Static Maps†

Eric Wemhoff and Andrew Packard
Department of Mechanical Engineering, University of California, Berkeley
{ericw,pack}@jagger.me.berkeley.edu

***Abstract*— We consider the estimation of unknown signals in structured models that are interconnections of known linear dynamic systems and unknown static maps, and contain unmeasured exogenous disturbances. A main motivation for considering this is a system identification problem in which such an interconnection exists, and the static maps are to be identified when the inputs and/or outputs of the maps themselves are not available. Our approach is to search for estimates of the unmeasured signals based on three main types of criteria, these being that they are consistent with the linear dynamic system, that stochastic assumptions for disturbance processes are met, and that input-output pairs of the static maps are consistent with there being a static relationship between them. We consider various candidate critera for enforcing the staticness consideration; they are essentially smoothness or regularizing criteria. These are what make our formulation different from other common estimation methods, for instance Kalman smoothing. We compare and constrast different methods using an example.**

I. Introduction

We work with models that are interconnections of two types of operators: linear (possibly dynamic) ones, and static (possibly nonlinear) ones. A canonical representation for such models is achieved by grouping the linear operators into a larger linear operator called $\mathcal{L}$ and the static ones into a larger static one called $\mathcal{S}$, as in Figure 1. The measured input output are u and y, while e represents unknown exogenous inputs or disturbances. Further, we assume the linear part $\mathcal{L}$ is a known quantity, while the static part $\mathcal{S}$ is not. The figure introduces signals z and w which are the input and output of the static part, and which may not be measured. The task we undertake is to estimate any unknown components of these.

A main motivation for considering this is a system identification problem in which such an interconnection and state of knowledge exists, where there static nonlinear maps to be identified and the inputs and/or outputs of the maps are not necessarily available. We further suppose that little is known about the static maps, in that they do not have a natural parameterization that is known or suggested from an analytical understanding of the underlying process.

Parameter estimation and its role in system identification is a well-studied subject (see for example [6]). Nonparametric elements in an interconnected model are therefore often "covered" using very general parametrized function expansions, followed by estimation of the parameters in those expansions. Expansions that are tried include neural networks, radial basis functions, Volterra kernels, and polynomials [1, 8].

† Supported in part by NASA, under Cal State Los Angeles subcontract.

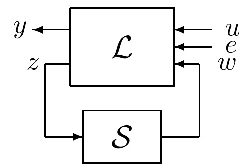


Fig. 1. Canonical model set

Sometimes this works well, and sometimes not. One issue is the large number of parameters often needed—numerical procedures can become intractable with respect to speed and convergence considerations. The cost functions involved also become very complex, which brings up the issue of choosing initial parameter estimates and basis functions. Poor basis function and/or initial parameter choice may likely result in divergence, or convergence to false local minima. On the other hand if there are too few degrees of freedom then estimates may be biased, due to undermodeling.

Now, working towards identifying the unknown static mappings of $\mathcal{S}$, suppose we conduct an experiment of length L. Assuming the data-generating system has the structure of Figure 1, it makes sense to talk about the signals z and w that were realized. If we had access to these then this of course provides information about $\mathcal{S}$. More specifically, for each t, the pair $(z(t), w(t))$ is an element of the graph of $\mathcal{S}$. In a sense the set of points$\{(z(t), w(t))\}_{t=1}^{L}$ is the most basic information about $\mathcal{S}$ contained in the experimental data; this is where the experiment sampled the function. However, some components of z and/or w may be inaccessible or difficult to measure, and so the problem becomes one of *estimating* these signals, so that we can then use their scatter plot as a nonparametric estimate of the nonlinearity.

This paper builds on earlier work ([10], [3], [9]). Here we extend and discuss different ways to formulate the main estimation problem, and basic issues involved. We also develop an example to illustrate ideas and compare other possibilities for doing signal estimation and system identification.

II. Solution Approach

Towards estimating unknown signals, we take the straightforward approach of stating desirable qualities the unknown signals should have, and performing optimization to find the best estimates. Perhaps the most evident information available is the input-output data. The linear system $\mathcal{L}$ defines a relationship between the signals u, e, w, y, z, namely a set of linear equality constraints. In estimating unknown signals we will require that they are consistent with the linear system and experiment data. This begins to narrow down the choices. These system constraints can be taken care of by adding them directly to optimization formulation. A more attractive option is to compute an explicit parametrization of this constraint set's feasible set and thus reformulate the problem with the constraints "built-in", rather than describing the set explicitly. This can work especially well with linear constraints in the setting of convex programming. It eliminates the constraint equations and reduces the number of decision variables, making for an easier optimization. In the present situation this is important, due to the large number of constraint equations and variables involved. Working in a discrete-time setting over a finite time interval, the system constraints take the form of a finite number of linear equations among a finite number of unknowns, and can be written as a matrix equation (involving Topelitz matrices of the system's state-space data). We assume this has at least one solution (for instance this is always true in the case of full measurement noise, in which case the unknown noise signal can simply be chosen to make the data consistent). The parametrization of the solutions to this equation can be computed using standard matrix operations, or ones that take advantage of the structure that derives from the linear system; Wemhoff (2003) has details about the latter approach, and code examples can be found at `http://jagger.me.berkeley.edu/~ericw/`.

We assume this is done and use the notation

$$\left\{ \begin{pmatrix} z \\ e \\ w \end{pmatrix} : \text{ consistent} \right\} = \left\{ \begin{pmatrix} z^f := z^0 + K_z f \\ e^f := e^0 + K_e f \\ w^f := w^0 + K_w f \end{pmatrix} : f \in \mathbf{R}^\kappa \right\}, \tag{1}$$

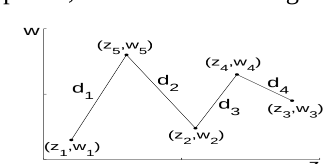
where z^0, e^0, w^0, K_z, K_e, and K_w are computed and reflect the input-output data, and e.g. z^0, K_z parametrize the set of consistent z signals. In the criteria and constraints we develop below unknown signals are replaced by their parametrized version (1). Thus f becomes the decision variable, and explicit linear system constraints are eliminated.

Another source of constraints and criteria comes from assumptions about $\mathcal{S}$. Foremost among these properties is that it is a static operator, and hence the partial graph of $\mathcal{S}$ implied by estimates of (z, w) should be congruous with this. Recall the definition: an operator is static if the output at time t depends only on the value of the input at time t. It does not depend on previous or future inputs, or time. In this case there should appear to be a single-valued function which relates input values to output values. This leads to the following preliminary idea. We do not allow estimates of the input and output of $\mathcal{S}$ for which a particular value of the input is repeated at two distinct times t_1 and t_2 i.e. $z(t_1) = z(t_2)$, but for which $w(t_1) \neq w(t_2)$. A static operator could not map such a z to such a w.

However this criterion turns out to not be especially useful in practice. The problem is that the set of finite data records with repeated inputs like this is very small; indeed when considered as a subset of all possible records of length L, it has zero measure. Even for an estimate which *is* ruled out by this criterion, there are others lying arbitrarily close (pick a norm) that aren't ruled out, which arise from small perturbations so that there are no longer repeated inputs. The modified input-output pair passes the test and is not at odds with a static relationship, and for all practical purposes it may be indistinguishable from one that does not pass the test. So we are able to rule out very few data records this way, and as for the rest—some of which are arbitrarily close to ones that are ruled out—one is just as good as another.

If a data record does *not* have repeated inputs then it is always possible to find a static operator relating the two. Simply fill in the missing points in the function's domain with any value. For example a polynomial can be found which interpolates the points. Another choice for scalar signals is to linearly interpolate between the points to form a static function. However, even though a particular finite-length input-output data record might not be technically inconsistent with a static input-output relationship, it is often the case that upon viewing the data one is able to intuitively make the assessment that a static relationship is unlikely. Loosely speaking our intuition is that the scatter plot of input-output pairs coming from a static operator should possess some minimal amount of smoothness. Here "smooth" does not necessarily refer to a definition in terms of differentiability, but a more intuitive notion that the scatter plot does not oscillate wildly and/or is not "cloudy". Further, this intuition becomes more reliable as the amount of data increases—scatter plots of nonstatic relationships tend to become more cloudy. Although this is certainly not the case for in general, for many static operators of practical interest this intuition is reasonable. Ultimately we need concrete quantitative cost functions that both capture our intuitive notions and are simple and conducive to use in numerical procedures.

Consider now SISO static operators, hence $(z(t), w(t)) \in \mathbf{R}^2$. A cost function that captures part of this intuition for smoothness, for a particular data record (z, w), is to sum the squares of the lengths in the $z - w$ plane of the lines connecting the points in the scatter plot, in order of increasing z. For instance consider the following scatter plot of a data record with 5 points, with the connecting lines drawn in



Here the cost is the sum of the squares of the four lengths d_i. For (z, w) a SISO data record of length L we thus define

$$\mathcal{J}(z, w) := \|\Gamma_z D^1 P^z z\|_2^2 + \|\Gamma_w D^1 P^z w\|_2^2. \tag{2}$$

We call this the *dispersion function*; here Γ_z, Γ_w are scalar weightings chosen by the user to adjust the criteria e.g. for the relative numerical sizes of the z and w signals. The name reflects the intent of measuring how widely dispersed the points are from lying on the curve of a static function, or how cloudy the scatter plot looks. The more cloudy, the larger the measure, while smoother graphs will result in smaller $\mathcal{J}$. Here P^z is the permutation operator which sorts the vector z in ascending order (a linear operator), and D^1 is the first-order difference operator defined by its operation on a vector v as $(D^1 v)(t) = v(t+1) - v(t)$ (also denoted $D^1_t v$).

We have found that often, choosing $\Gamma_z = 0$, i.e. neglecting the contribution from the first term in (2), does not degrade estimates too much, and results in simpler formulations. Another variation on $\mathcal{J}$ is to replace first-order differences with second-order ones, resulting in estimates that reflect the higher-order smoothness penalty. Another variation is to sum the lengths (instead of the squared lengths):

$$\mathcal{J}_{\mathrm{A}}(z,w) := \sum_{t=1}^{L-1} \left((\Gamma_z D^1_t P^z z)^2 + (\Gamma_w D^1_t P^z w)^2 \right)^{1/2}. \quad (3)$$

With $\Gamma_z{=}\Gamma_w{=}1$ ($\Gamma_z{=}0, \Gamma_w{=}1$) this is the arclength (total variation) of the connected scatter plot. $\mathcal{J}_{\mathrm{A}}$ still enforces the intuitive smoothness, but penalizes large jumps less than $\mathcal{J}$ (and so may be a better measure to use for unknown maps which may contain discontinuities). It is sometimes possible to know the total variation of an unknown map and thus constrain it, in contrast to dispersion where it is usually the case than an appropriate constraint level is not known, and thus this terms is more appropriately used as an objective rather than a constraint.

A third important source of estimation criteria is known or assumed statistical properties of the disturbance signal e, such as its mean and variance, autocovariance and spectrum, and correlation with other signals (or lack thereof). Most fundamentally, we would like any estimate for e to be a likely sample path according to the assumed stochastic model. By limiting the realizations of the noise that will be considered or preferring some to others, this limits the estimates for z and w to be considered or makes some more preferred than others, via other constraints and criteria that relate e to z and w. It is computationally straightforward to constrain the mean and two-norm of the estimate of e, as well as correlation with signals in the problem that are measured; these are all convex quadratic functions of the decision variables f. It is more difficult to form computationally reasonable criteria for autocovariance, and correlation with unknown signals. As the two-norm can be thought of as a statistical estimator of the variance, this simple but useful criterion tends to enforce that the estimated signal be consistent with a stochastic process of a certain variance.

These three criteria form the building blocks. In order to be able to solve problems and not just pose them we place some emphasis on working within a framework that allows use of efficient solvers that can handle large problem sizes, since our decision vectors are multiples of the data record length, and thus can be large. So we concentrate on the convex cone family of problems, which can efficiently handle the large problem sizes involved, and still remain fairly flexible. In this paper the problems we ultimately hand off to solvers are quadratically constrained quadratic programming (QCQP).

III. Choices

Whether or not the z signal is known has computational implications (whether w is known is not as important). Each of the staticness objective and constraint functions discussed above are nonconvex due to the way in which they depend on z, and hence are nonconvex in f. When z is known or assumed then we can use that fixed value in the problem parts of the criteria/constraints, and the remaining dependence on f is well-behaved and convex. Otherwise, when z is unknown, $\mathcal{J}(z,w)$ and friends are discontinuous, nonconvex, nonquadratic, rather messy functions that are not handled by solvers that are workable for the length of the decision vectors we want to consider. This is primarily due to the sort operation, which we recall is needed in order to put the data record in order of increasing z, so that we can take the appropriate differences. Therefore approximations are needed for the problem we would ideally like to solve. We can suggest three options in this situation:

1. Settle for a formulation using only criteria/constraints which are "nice" functions of f. This reduces the available options. One that is still available here can be interpreted as a Kalman smoothing operation, or as an ill-posed inversion problem with a regularization criteria.
2. Bootstrap. This is how we refer to an iterative process in which each iteration produces new estimates of the unknowns e, w, and z. To proceed, those parts of the criteria/constraints that cause problems (usually this is the sort operation) are fixed at the current iteration's estimate of z, leaving an optimization problem which can be formulated as a convex cone problem. This is then solved, producing estimates for the next iteration. We have found this often works reasonably well, and is what we generally try first when z is unknown.
3. Reformulate the problem as one that is not quite a convex cone problem, but can be *relaxed* into one. To date we have tried reformulations that involve only criteria/constraints that are quadratic, but possibly non-convex. For these problems the S-procedure (Boyd et al. 1994) may be employed to produce bounds on the optimal value, and quantities produced in solving this problem can be used in a guided search for suboptimal feasible solutions to the original problem (Goemans and Williamson 1995). The size of the optimization problem grows and so here we're restricted to shorter data record lengths, and once again the set of available problem formulations is reduced since all criteria/constraints must be quadratic, however initial trials have shown promise.

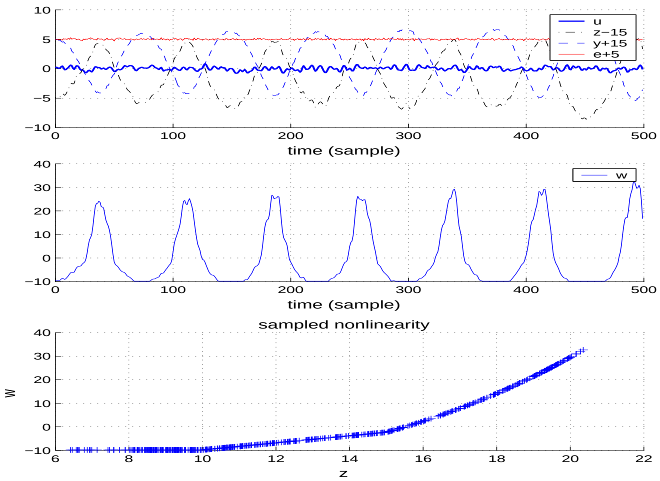


Fig. 2. bungee example: Simulation data.

IV. Example

For illustration we use a simple mechanical example. Consider a mass m suspended by an elastic cord from a point whose vertical position u we can control (upwards is the positive direction). A noisy measurement $y = x + e$ of the vertical position x of the mass is available. The noise is additive IID zero mean normal with variance σ_e^2. The model includes a gravity force $F_g = -mg$, viscous damping c on the mass, and a restoring force due to stretch in the cord. The restoring force $F_r = k(u - x)$ is a nonlinear function of the length $u - x$ of the cord; when the cord is stretched it acts as a hardening spring after an initial linear elasticity regime, and when the cord is slack it provides no restoring force. Discrete-time equations of motion come about from discretizing the second-order continuous-time ODE using an Euler method with sample time $T = .05$ seconds. They are

$$x(t+T) = x(t) + Tv(t) \tag{4}$$
$$v(t+T) = v(t) + T\left[-cv(t) + w(t)\right] \tag{5}$$
$$y(t) = x(t) + e(t) \tag{6}$$
$$w(t) = k(z(t)) - g =: k_e(z(t)) \tag{7}$$
$$z(t) = u(t) - x(t), \tag{8}$$

where z and w are the input and output of $k_e(\cdot)$, respectively, and the "effective" restoring force $k_e(\cdot)$ accounts for both gravity and cord stiffness. Note that z is nearly measured, since $z - e = u - x - e = u - y$. However the restoring force w is not measured; the main problem is to estimate this quantity. All signals here are scalar.

Input-output data is simulated with lowpass-filtered uniformly distributed white noise u and gaussian white noise disturbance e. Figure 2 shows 500 samples of typical simulation data with $\sigma_e = .1$. The most negative w ever gets is -9.8, when the cord is slack. The last panel is a scatter plot of w versus z; this is where the nonlinearity $w = k_e(z)$ was sampled in this data record. It shows the shape of the nonlinearity; on it can be found the cord length when taut but not stretched (10), the transition from linear to cubic hardening (15), and the offset force due to gravity (-9.8).

In this section we essentially concentrate on following estimation choice: minimize $\mathcal{J}$ while constraining $\|e^f\|_2$ to some value near an assumed variance. This formulation is among the simplest that makes use of one of our staticness critera, and has been observed to gracefully accommodate a wide variety of situations in practice. For input-output data we use the first L steps of simulation data, where the simulation is fixed and L varies in order to explore the effect of the amount of data. Using this data along with knowledge of the linear system, the next step is to compute a parametrization of the consistent solutions (e^f, w^f, z^f), with f the free parameter. As z^f in this problem is indeed a function of f, we apply the bootstrap method. At iteration k fix $P = P^{z^{f^{(k)}}}$, and solve

$$f^{(k+1)} = \begin{array}{cl} \underset{f}{\text{argmin}} & \|D^1 P w^f\|_2 \\ \text{subject to} & \frac{1}{L}\|e^f\|_2^2 \leq \alpha\, \sigma_e^2. \end{array} \tag{9}$$

Here α is a design variable used to choose the noise constraint relative to the actual simulation noise variance σ_e^2. To compute solutions we use SeDuMi [7], an efficient solver for convex self-dual cone problems, of which (9), being QCQP, is an example. We chose to examine 2 noise levels, $\sigma_e \in \{0.1, 1\}$, and three data record lengths, $L \in \{100, 500, 1000\}$. Only the noise level was varied, the other details of the simulation are the same for all 9 cases.

We chose $\alpha = 0.95$ (experience has shown that a value slightly less than unity is generally a good choice) and $f^{(0)}$=0. In this example the second bootstrap iterate is much improved over the first, and later iterates produce smaller changes. The iteration is generally seen to improve the overall estimate, but not always uniformly over the domain of the function, and some portions of the domain may be estimated more poorly than in the previous iteration. Our experience has been that the sequence of estimates produced by the iteration is not monotonic in (pick a norm, e.g. $\|w - \hat{w}\|$) and in later iterates ends up "bouncing around", most likely due to the highly discontinuous re-sorting operation involved. Nevertheless the iteration is useful in that it improves initial estimates and these later oscillations are relatively smaller than the initial steps of more substantial improvement.

Figure 3 shows the quality of the estimated w–z scatter plot for the six combinations of σ_e and L. First of all, these indicate that the chosen optimization formulation is doing a reasonable thing. From the estimates, and especially from the better ones, we get a fairly clear picture of the shape of the nonlinearity. As we should expect the estimation quality gets better as noise level decreases, a reasonable thing to ask. At $L = 100$ the estimate quality is already quite good for the low noise levels. Figure 4 compares a portion of the estimated signals with the simulation data (in the bottom panel we plot the $\sigma_e = 0.1$ noise, and scale the $\sigma_e = 1$ estimate down by

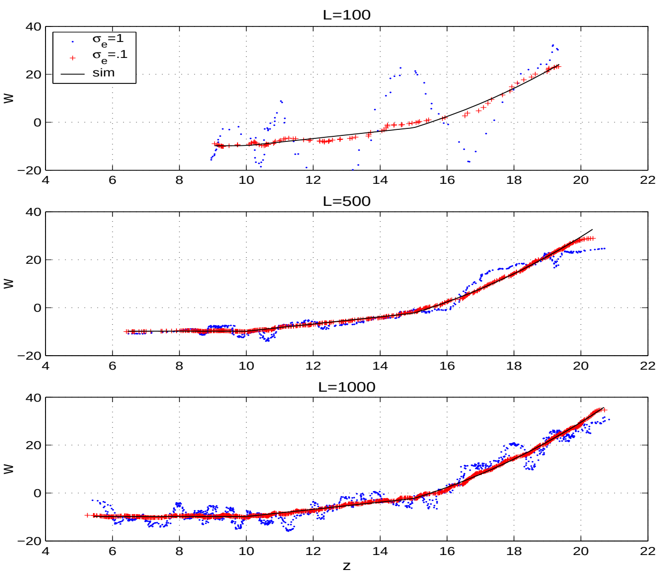


Fig. 3. bungee example: Scatterplots of estimated w and z, for different amount of the data record.

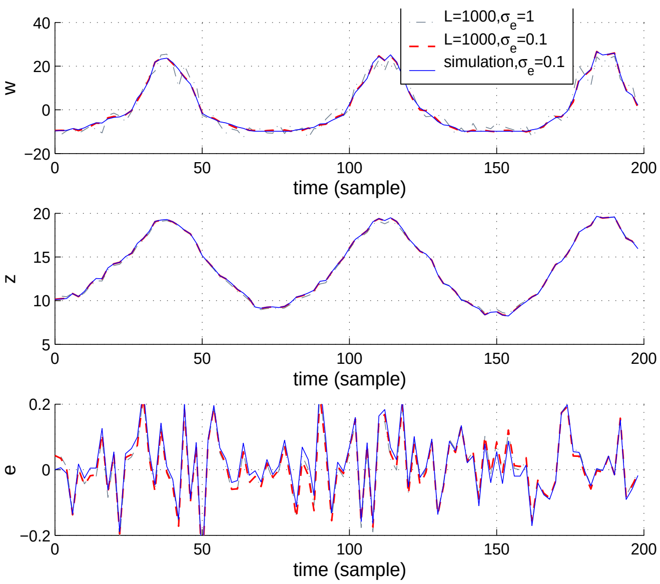


Fig. 4. bungee example: Comparison of estimated and true w, z, and e.

10 for purposes of comparison). The estimate corresponding to $\sigma_e = .1$ tracks the simulation signals quite well, and with more noise the estimate degrades.

It is interesting to compare estimates made using a simpler criteria, as a check that the dispersion measure is valuable. Consider a criteria which is, very simply, the 2-norm of w. Using this in place of $\mathcal{J}$ the optimization problem becomes

$$f = \begin{array}{ll} \underset{f}{\text{argmin}} & \|w^f\|_2 \\ \text{subject to} & \frac{1}{L}\|e^f\|_2^2 \leq \alpha\, \sigma_e^2. \end{array} \quad (10)$$

Here the estimation of (e, w) and z is decoupled because the constraints and criteria do not depend on z. The problem is immediately convex and quadratic in the decision variables, and no bootstrapping is necessary. In the remainder of this section we use 500 points of the $\sigma_e = 0.1$ simulation data.

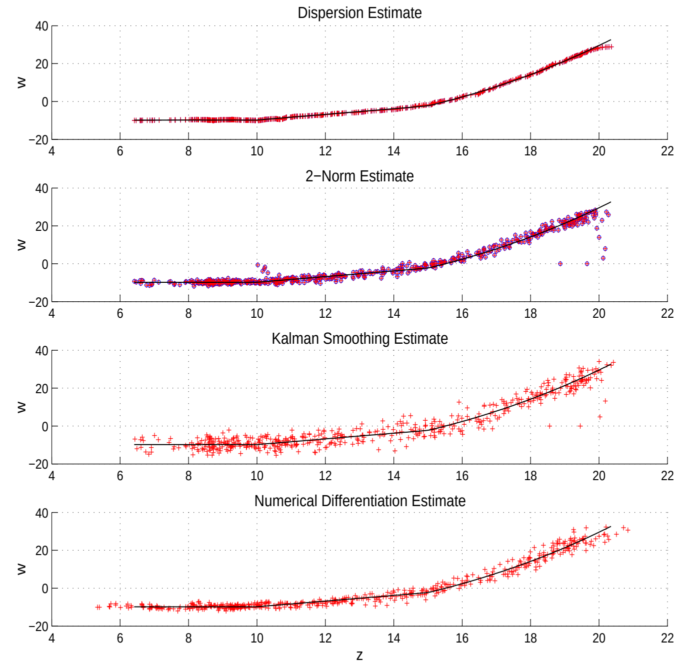


Fig. 5. bungee example: Comparing the dispersion, 2-norm, Kalman smoothing, and numerical derivative estimates. $L = 500$, $\sigma_e = 0.1$.

The scatter plot of the estimates produced by this approach (again α=0.95) are shown in the second panel Figure 5, along with the previously-seen estimates using $\mathcal{J}$ in the top panel. Again results are reasonable, in that the trends are certainly right and the estimates appear unbiased. But the variation of the estimated scatterplot about the true, simulated one is much larger than when using the dispersion measure. When we check, the 2-norm of $\hat{w}$ is indeed (slightly) smaller than before, however $\mathcal{J}(\hat{z}, \hat{w})$ is much larger.

Perhaps the benchmark when it comes to problems of signal estimation is Kalman filtering, or in our offline setting, Kalman smoothing (KS). Its contribution (besides using a computationally efficient iterative algorithm) is to quantify the optimal tradeoff between trying to minimize both $\|e\|$ and $\|w\|$ when these signals are independent white Gaussian signals of known variance (and of course neglecting that the relationship between z and w is static). Taking advantage of the fact that we know the correct variance of e and w in our simulated data record we compute their KS estimates, compute $\hat{z} = L_z(u,\, \hat{e},\, \hat{w})^T$, and plot in the third panel of Figure 5. The results are similar as those for the 2-norm criterion (but with higher variance). This is actually not surprising. The KS solves

$$(\hat{e}, \hat{w}) = \underset{e,w}{\text{argmin}}\, \|e + \gamma w\|_2 \quad \text{s.t.} \quad y = L_y\left(u, e, w\right)^T, \quad (11)$$

with γ chosen according to the relative sizes of e and w, and (10), with the system constraints made explicit again, is like

$$(\hat{e}, \hat{w}) = \underset{e,w}{\text{argmin}}\, \|w\|_2 \quad \text{s.t.} \quad y = L_y(u, e, w)^T, \|e\|_2 \leq \alpha' \quad (12)$$

It is a fact that for a solution of (11), there exists an α' such that (12) produces the same solution, and conversely for a solution of (12) there exists such a γ. Here, when running KS with a variance level for w around $0.05\sigma_w^2$, where σ_w^2 is the actual variance, the KS and 2-norm results are identical. Comparing (9)–(12) we get a good idea of how the dispersion estimate formulation differs from Kalman smoothing.

We can compare these results with a more ad-hoc approach. Looking at the problem formulation again we see that w actually represents the acceleration of the mass. Since a position measurement is available why not 1) use that to compute the stretch of the cord i.e. z and 2) differentiate twice to compute the acceleration of the mass i.e. w. Since the position measurement is noisy, to give this a fair shot the first thing we did was to filter out higher frequency noise. We used a 3rd-order Butterworth with cutoff at 5% of the sample frequency, which seems to produce about the best results from among several choices of order and bandwidth. The scatter plot of the estimates produced in this way is shown in the fourth panel of Figure 5. Again the dispersion estimates are considerably better than this more ad-hoc idea.

As mentioned in the introduction, when dealing with unknown nonparametrized nonlinearities, a common idea employed in the system identification field is to use a finite parametrization which can represent a large function class, and estimate the parameters involved. We can comment on our experience doing this for the bungee example. A parametrization was chosen in which k_e is represented as a linear interpolation between 40 fixed points in its domain, the value of the function estimate at those 40 points being the parameters ("triangle" basis functions). We use a maximum-likelihood formulation, which for our problem with Gaussian measurement noise, is equivalent to minimizing the norm of the residual between the measured output, and the simulated output at the parameter value and with no noise. To solve we use the nonlinear programming (NP) package NPSOL (Gill et al. 1994). Once again we use the $L = 500$, $\sigma_e = 0.1$ data set. We found that for reasonable starting conditions, NPSOL is often unable to converge to a good solution, apparently due to numerical difficulty. On the other hand the nonparametric methods presented here appear to be more robust to their initial conditions, and usually converge to a reasonable approximation of the graph independently of the choice of $f^{(0)}$. But, they have difficulty handling long data records, and indeed for some examples more data does not seem to improve estimates. What does work, however, is to use the nonparametric estimate to form more accurate initial parameter guesses in the parameter estimation formulation. In this example, we fitted initial parameters to nonparametric estimate in e.g. Figure 3, and the NP solver was then able to converge nicely to an improved estimate. Thus these ideas for nonparametric estimation of the graph of a static function may be useful as a complement to parameter estimation methods.

V. CONCLUSION

We show ideas for estimating unknown signals from input-output data when using a model that involves unknown static maps. The dispersion criteria is shown to be a useful tool here. It produces reasonable results, and its performance was compared to several other approaches including Kalman smoothing. The dispersion measure is one of a related family of staticness or smoothness criteria that might be used. Extensions are available for 1) situations where there is greater than one nonlinearity to be identified (and hence several independent staticness relationships are present) and 2) nonlinearities with more than one input or output. For MIMO $\mathcal{S}$ with input-output structure in which some outputs do not depend on some inputs, it is important to take that into account. It is also possible to take advantage of knowledge of elements that are repeated or related or have known smoothness properties. Future work involves exploring the advantages and disadvantages of using these to craft estimates. More work is needed in exploring the bootstrap iteration and other ideas for handling the situation when z is not measured, including convex relaxations of harder problems. Another important generalization is to be able to handle models that involve *known* nonlinearities; in general this leads to a hard nonlinear optimization.

REFERENCES

[1] S. Billings. Identification of nonlinear systems–a survey. 127(6):272–85, Nov. 1980.

[2] Boyd et al. *LMIs in system and control theory*. SIAM studies in appl. math., v. 15, 1994.

[3] M. Claassen. *System Identification for Structured Nonlinear Systems*. PhD thesis, Univ. Calif., Berkeley, 2001.

[4] Gill et al. User's Guide For NPSOL 5.0: A Fortran Package for Nonl. Prog. Tech. rep. Stanford Univ. 1994.

[5] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinate programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[6] L. Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, 1987.

[7] SeDuMi. A Matlab Toolbox for Optimization Over Symmetric Cones. by Jos F. Sturm.

[8] Sjöberg et al. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12): 1691–1724, 1995.

[9] E. Wemhoff. *Signal Estimation in Structured Nonlinear Systems with Unknown Functions*. PhD thesis, Univ. Calif., Berkeley, 2003.

[10] E. Wemhoff, A. Packard, and K. Poolla. On the identification of nonlinear maps in a general interconnected system. In *Proc 1999 ACC*, v. 5, pp. 3456–61, 1999.