Ryan Decker – Computer Integrated Surgery II – Critical Review
March 5th, 2012

Introduction

This critical review will analyze an important paper in the area of systems identification and relate its insight to the specific engineering task of developing a better kinematic model of the Revolving Needle Driver (RND) robot, specifically its Remote Center of Motion (RCM) module. The Remote Center of Motion is a concept central to laparoscopic surgery. The idea is that an end effector can be constrained at a point which is not directly attached to the robot. This is useful in needle insertion, where the insertion point should remain fixed while the needle probes a conical volume inside the skin. A well-known example of this approach is the DaVinci medical robot, which performs laparoscopic surgery through remote control with a constrained parallelogram mechanism. While some movement of the RCM point is allowable in less dextrous clinical applications, more precise RCM modules are needed in for ocular surgery and other micro-surgical fields. This RCM is a common stage and is present in several robots designed by the URobotics Laboratory. It currently has some error due to unknown imperfections which arose during construction. With knowledge of the **inputs** and **outputs** to this system, we must develop a new model which will explain its behavior.

Currently the end effector (needle tip) position is modeled as a function of three inputs. These are RCM angle 1 (the 'base' or 'X' angle), RCM angle 2 (the 'Y' angle), and needle insertion depth. The RCM angles both have constant offsets. Our task is to observe tip positions at an array of known joint angles and fixed depth with a Polaris optical tracker, and compare with the expected tip placement. Knowing the errors, we can determine if the robot has a high precision and a low accuracy, which could be fixed by a calibration of some parameters (link length, joint angles, etc), or if it has low precision and high accuracy, in which case the kinematic model will need to be rewritten using new basis functions. This process will iterate until we converge on a model with more reasonable error.

Paper - "Nonlinear Black-box Modeling in System Identification – a Unified Overview"

The paper in question was written by Jonas Sjoberg et al in 1995. Chosen for its breadth and depth, it is meant to serve as an introduction to systems identification and parameter estimation for all branches of applied mathematics including engineering, biology, medicine, and computer science. Written from a statistical perspective, it may take some adaptation of mathematics to apply its insight, and we will use only the most relevant information. The examples given are of a higher mathematical abstraction than is necessary for many purposes, but serve as a good overview. It also contains a great blueprint for systems identification which can be applied to the link parameter problem in the RND robot.

We start with a categorization of the system which we are trying to model. The authors refer to three cases: white, grey and black box systems. White signifies that all contributing factors are known from prior knowledge and physical insight. Grey-box models are applicable when some parameters are known, but others must be determined experimentally. This is the case which interests us. It occurs when a model (kinematic or otherwise) has some free undetermined parameters. The sub-case especially applicable is that of physical modeling, which will hopefully explain error in the RCM module. It is more likely that the combination of many errors from the RCM and small errors in the optical tracker will nudge the project towards the realm of semiphysical modeling, where some insight from geometric information can be subjected to black-box techniques, which the authors explain in depth.

Black-box modeling is preformed when no prior model of the system is known. All that is available are the observed inputs and outputs, and what is desired is a relationship between past I/O data and future outputs.

Inputs and Outputs

$$u^t = [u(1) \quad u(2) \quad \dots \quad u(t)],$$

$$y^t = [y(1) \quad y(2) \quad \dots \quad y(t)].$$

Looking for relationship

$$y(t) = g(u^{t-1}, y^{t-1}) + v(t).$$

This relationship should contain a finite number of parameters (θ) which will typically correspond to the scale and location of relationship support, and restrict the function to a family of mathematical structures. This family of model structures is still too broad, so it is written as a function of: 'parameters' and a nested 'function of past I/O pairs'. The nested function of past I/O is called the regression vector (φ) and can itself be parametrized. There is great research into the types of regressors which will model certain situations well, and they can become arbitrarily complex as the number of parameters grows. The authors describe several nonlinear models derived from varying regression functions.

$$g(u^{t-1}, y^{t-1}, \theta) = g(\varphi(t), \theta),$$

where

$$\varphi(t) = \varphi(u^{t-1}, y^{t-1}).$$

These parameters and regressors are part of nonlinear mappings (g) from the regression space to output space. One can conceive these mappings as a family of function expansions with scaled basis functions (g_k) of regressors. The parameters and basis functions provide a good framework for most black-box modeling structures.

$$g(\varphi, \theta),$$

which for any given θ goes from $\mathbb{R}^d$ to $\mathbb{R}^p$.

In fact, many nonlinear black-box systems can be described by parameterizing a single 'mother basis function' called κ(x). This mother basis function will be a function of the regression vectors, a directional scaling term, and a translational (offset) term. The authors provide multiple examples differentiating specific structures of these basis functions and parameters.

$$g_k(\varphi) = \kappa(\varphi, \beta_k, \gamma_k) \quad \text{`}= \kappa(\beta_k(\varphi - \gamma_k))\text{'}.$$

Individual basis functions must be built from the mother basis function. Single variable basis functions can be classified as global or local, depending on their range of support, while multivariable basis functions require more advanced techniques for their construction which may provide a better fit in multiple directions. The authors show construction of the mother basis function in three ways, each giving different directional support in output space. The first gives radial support which decays as distance from the offset grows. The second gives semiglobal support focused on a function in regression space, which can be visualized as unbounded support along a ridge. The third is the construction of a tensor product which can give varying support in different directions, but should not be used in high-dimensional cases due to high computational cost. Before we estimate too many variables, it is important to stop and look at what information is already available.

One should not estimate what is already known. Parametrization of measured geometries will lead to unstable or overconstrained models. So it is useful to connect the system under scrutiny to some similar, more determined system. The authors show this by defining model quality as the difference from the 'true' model g_0, which is observed data in most cases. The difference contains noise as well as two important parts: bias and variance. Bias is the difference between the best model in this family θ_m (of dimension m) and the true model. Variance is the difference between the current model θ_N (with N regressor-output pairs) and the best model.

$$
\begin{aligned}
V_*(m) &= E\bar{V}(\hat{\theta}_N) \\
&= \lambda + E\,\|g_0(\varphi(t)) - g(\varphi(t), \hat{\theta}_N)\|^2 \\
&\approx \underbrace{\lambda}_{\text{noise}} + \underbrace{E\,\|g_0(\varphi) - g(\varphi, \theta_*(m))\|^2}_{\text{bias}} \\
&\quad + \underbrace{E\,\|g(\varphi, \theta_*(m)) - g(\varphi, \hat{\theta}_N)\|^2}_{\text{variance}}.
\end{aligned}
$$

So it is with low variance and easily observable bias that we may find a better model. If the best model is known, then adding more parameters will decrease the bias. However, if the best model is estimated, adding more parameters may decrease overall model quality if the reduction in bias is less than expected variance. So it is wise to consider the family of models best suited for each system and not to add parameters 'spuriously'. Here model quality is stated in terms of bias and variance:

True model

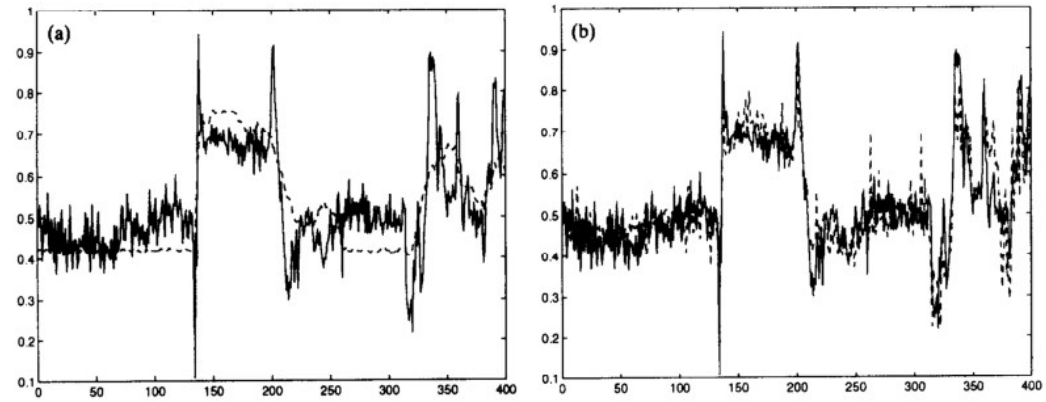
$$y(t) = g_0(\varphi(t)) + e(t)$$

Quality

$$
\begin{aligned}
\bar{V}(\theta) &= E\,\|y(t) - g(\varphi(t), \theta)\|^2 \\
&= \lambda + E\,\|g_0(\varphi(t)) - g(\varphi(t), \theta)\|^2
\end{aligned}
$$

The authors discuss several methods to tease out the parameters which are most important for a given model structure. One such method is called regularization, which attaches a penalty to a parameter to discover those which affect the model quality most. Another idea is to omit basis functions selectively, perhaps removing those below a threshold. For all methods presented, the idea is to iterate parameter selection processes with more expected data sets and to 'train' the model to fit the data.

With each training iteration, the number N of regressor-output pairs grows. If this process is carried out, the model will be overwhelmed with parameters, and subsequent decreases in bias will be dominated by higher variance costs. This is the tradeoff which results in an 'efficient' number of parameters.



This is the general process of black-box systems identification. The authors spend the remainder of their time confronting structural issues in specific model families by presenting key examples, none of which are directly applicable to the complex task of RCM module parameter estimation. In the example below (patient's glucose levels), the original function is modeled by the dashed line. The model on the left is generated from initial patient data and regression model, and the right is generated by an optimization which learns from repeated trials. This optimization clearly generates a better model, and it is what we will need to do in our limited case of the RND.

Assessment

Although no novel methods are presented, the overview is given using a common framework and serves the interests of the user well. It is significant that the authors could complete such a broad overview of the field by using relatively simple algorithmic steps which are generalizable to a large number of systems. The statistical approach encompasses a wider net of applications and succinctly explains a difficult and unknown problem. The accompanying techniques vary widely in their complexity, showing that the results of a systems identification often depend on analysis depth determined by the identifier. The paper is organized well. First a general approach to systems identification is given, which is accompanied by classes of regressors and basis functions previously established. Then several examples show the effectiveness and weaknesses of these methods.

This paper is an important tool for those confronted with a systems identification problem for which there is little other prior art. When a novel system must be identified, the general steps outlined in this paper will guide an analysis which can be arbitrarily complex. The range of applications presented drives a high citation rate in numerous fields. It is thorough and broad, but what can we use for the link parameter estimation of an RCM?

The paper contains examples applicable to everything from neural networks to DC motors. It describes many high-level mathematical techniques for explaining the 'error' in systems by identifying the structure and parameters of an enhanced model. However, this desire to be broadly applicable leads to such a diverse array of examples that the paper can seem unfocused and irrelevant at times. There is a trade off between universality and extraneous information, much like the trade off between variance and bias. If too much information is included, the user/model will be overwhelmed. Therefore it is desirable to pick portions of the approaches described which will work for the current situation. Some topics explored are not relevant to our modeling of the RND robot. For example, the paper incorporates fuzzy logic, the notion that reasoning can be approximate, and a Boolean value may have a value between 0 and 1. We will not look at the fuzzification of link parameters directly, but we will group them into larger sets of parameters which may be easier to estimate in a simplified system. Although it is stated that System Identification cannot be fully formalized or automated, the summary at the end of the paper provides a great guide for anyone performing systems identification, and will be broken down in the following paragraphs and related to the project task.

The fist step in system identification is to look at the data output by the system. It can be assumed the input variables are known fairly accurately. Since the output does not match the expected result, there must be some error in the output data. By observing the errors, one may be able to determine possible causes. For example, does the error change proportionally to an input joint angle? In what directions is the error growing? Are there certain regions of the robotic workspace which experience noticeable changes in error? Does the error change with time? These questions are best answered by collecting output data from single inputs. In the RND case, RCM error could be observed by actuating both rotary joints independently at all possible configurations and observing the needle tip position for a fixed insertion depth. RCM error is clearly a function of the three input parameters previously mentioned, but correction will focus on the RCM module which actuates the two rotation angles. The RND robot was selected for this project due to its precise control over needle insertion depth.

After observing modeling errors, correction by systems identification should start with the simplest explanation. This may not account for all observed error, but will serve as a first step for the process of iteration. By starting simply, one may find physical insights or linear models to explain the error, which is highly desirable. Our analysis of the RCM module should start as a grey-box model, using a mechanical dissection to inform 'known' parameters. Estimation of a more correct model will begin from this idealized version. For example, an additional parameter may be an offset angle not previously recognized. A misalignment of the RCM module links may necessitate basis function expansion to rotate about the misaligned axis. As the number of parameters increases, the complexity of the model grows and becomes harder to estimate with successive iterations. It should always be a goal, especially of the engineer, to look for parametrization with physical justification. The authors emphasize this even throughout discussion of black-box techniques, where physical insight is typically not available. It is important to look for these intuitive explanations because they can explain the I/O relationship with fewer parameters than arbitrary basis functions, leading to a higher quality model faster.

Quality criterion for parameter estimation can be deconstructed into two parts, bias and variance. When collecting data, it is desirable to have low variance in order to observe, and correct for, more bias. It is the bias which informs the parameters, while variance can be characterized with an error term. There can be an increase in true approximation with an increase in the number of basis functions.

However, with more basis functions comes more variance. Ideally, there should be a minimization of bias with the use of a few basis functions. This requires the basis functions to be efficient. Smart modeling will use an efficient number of basis functions which accept a limited number of parameters with great effect. For this, the observed bias should be as pronounced as possible. It is necessary for us to tease out much variance, as the bias which we wish to observe is very small, in the millimeter range. Therefore noise must be reduced at all stages of observation. This includes noise in encoder count/joint angle ratios, noise in the optical tracker, noise due to room vibrations, and other sources of extraneous variance which will cloud the observation of model bias.

Future research in this field should focus on the identification of model families (of basis functions) which work well for certain applications. This will hopefully move the modeling of more systems from the black-box class to the grey or white-box classes. A general framework has been laid out here with relevance to many specific cases, so the majority of work in the field will relate to expanding the quality and number of models which fit novel data well.

Future research for our project should focus on grey-box systems identification, as we already know some of the model parameters and what basis functions would be appropriate. These are link parameters and joint angles informing orthonormal rotation matrices. Hopefully we will be able to identify linear basis functions with physically intuitive parameters. However, it is beneficial to start with the most difficult case of systems identification and then make assumptions, this is why reading starts with the nonlinear black-box case.

Application

For our purposes, the grey-box case is applicable. We know some forward kinematics of the RND robot, and would like to optimize a few parameters.

The RND robot uses 2 rotations and 1 translation to take the needle tip to the world frame which is fixed at the RCM. This transformation can be discribed by an Euler angle formulation with 3 rotations about the axes, and the same translation. We can also add offset terms to any variable. Here we will add them to the needle depth, Rx angle, and Ry angle. This is shown in the following MATLAB excerpt. Note that the addition of these parameters causes the equation to be cut off.

```
%Now here are the rotation matricies to take tip to XYZ robot space (world)
phi0 = deg2rad(Rx); %RCM angle about X axis
phi1 = deg2rad(Rz); %RCM angle about Z axis
d = Ty; %Needle depth
World = []; %container for the ideal XYZ coordinates
Transformations = [];
for i = 1:length(Rx)
Rd =[1 0 0 0;
    0, 1, 0, -d(i);
    0, 0, 1, 0;
    0, 0, 0, 1];
Rphi0 = [1 0 0 0;
        0 cos(-phi0(i)+alpha) -sin(-phi0(i)+alpha) 0;
        0 sin(-phi0(i)+alpha) cos(-phi0(i)+alpha) 0;
        0 0 0 1];
Rphi1 = [cos(phi1(i)) -sin(phi1(i)) 0 0 ;
        sin(phi1(i)) cos(phi1(i)) 0 0 ;
        0 0 1 0;
        0 0 0 1];
%Euler angle formulation: No beta/alpha
Euler = [cos(phi1(i)), -sin(phi1(i))*cos(phi0(i)), sin(phi1(i))*sin(phi0(i)), 0;
        sin(phi1(i)), cos(phi1(i))*cos(phi0(i)), -cos(phi1(i))*sin(phi0(i)), 0;
        0, sin(phi0(i)), cos(phi0(i)), 0;
        0,0,0,1];
%With beta and alpha
Euler2 = [cos(phi1(i))*cos(beta),cos(phi1(i))*sin(beta)*sin(phi0(i)+alpha)-sin(phi1(i)
        sin(phi1(i))*cos(beta), sin(phi1(i))*sin(beta)*sin(phi0(i)+alpha)+cos(phi1(i))
        -sin(beta), cos(beta)*sin(phi0(i)), cos(beta)*cos(phi0(i)), 0;
        0,0,0,1];
World = [World; (Euler2* Rd* [0;0;0;1])'];
end
```

This is our relationship between the inputs (angles) and the outputs (XYZ) coordinates. It is a transformation matrix with mostly sinusoidal regression functions. The procedure for grey-box estimation is simpler than the black-box case, as we can guess and add parameters with physical intuition. But the advice of the black-box case holds generally. When these three parameters are optimized and learn over iterations, the regression function will more closely match the real world. I have optimized these parameters by training them to a data set of XYZ coordinates recorded by a Polaris optical tracker, and attached the results:

Targeting Accuracy (mm)	Before Optimization	After Optimization
	1.88	1.14

This accuracy is a mean norm error between the model and observed. Optimization of these parameters helps take care of many other errors which were not modeled. It increases accuracy beyond the few mechanical parameters which were available for adjustment on the robot.

Conclusions

This paper has great value to many fields, and lays out a useful framework for our purposes, even if a large portion of content is not relevant. It directly addresses the questions which arise during systems identification and tries to guide user choices towards a successful application. It is left to the user to research specific mathematical techniques most applicable to their situation, as is the case for much of systems identification. It is a great starting point for estimation of the unknown.

Source

Sjöberg, Jonas. "Nonlinear Black-box Modeling in System Identification: a Unified Overview." *Automatica*. 31.12 (1995): 1691-1724. Print.