Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Sensor Undistortion Using Hyperbolic Splines in Least Squares Sense

YangQuan Chen, *Senior Member, IEEE* and Yashodhan Tarte, *Student Member, IEEE*

***Abstract*— This paper focuses on the removal of static nonlinearity from a set of monotone increasing input-output data points, e.g. from a sensor. Möbius transformations are used to fit a spline to the data points to ensure smoothness and analytic invertibility of the interpolant. A new method is proposed to identify the coefficients of the Möbius transformation so as to minimize the twistiness of the resulting interpolant in a least-squares sense. Simulation results are provided to demonstrate the effectiveness of the method. Performance of the method in the presence of observation noise is also investigated.**

***Index Terms*— Hyperbolic splines, invertibility, Möbius transformations, nonlinear distortion, observation noise, sensor linearization.**

I. INTRODUCTION

In various applications such as signal and image processing, control systems etc., sensors form an integral part of the system. The main purpose of a sensor is to generate a signal proportional to the physical quantity to be measured. But in almost all cases, the input-output characteristics of sensors exhibit nonidealities like offset, gain, and nonlinearity, in addition to the possible effects of a disturbing variable which affects the desired sensor response to the physical quantity of interest [1]. This nonideal behavior affects the overall performance of the system.

The techniques used for detecting nonlinearities include using Hammerstein models (instantaneous nonlinear distortion followed by an all pole filter) ([2], [3]), higher-order statistics tools ([4], [5]), change in probability density function of the signal [6] etc.

The most straightforward method to compensate for the nonlinearities, as suggested in [7], is to pass the output of the nonlinear system through its inverse. Other methods include using artificial neural networks [9], adaptive filtering [3], [10] etc. In [11], piece-wise orthogonal polynomials are used as the "linearizer" for the hard-disk read/write head linearization process.

We focus on splines of the form $x \mapsto (ax+b)/(cx+d)$, as proposed in [7], to fit to the monotonically increasing data points

$$D = \{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$$

obtained by sampling the function $y = f(x)$ at several places. We want to find a function $\hat{f_D}\colon [x_0, x_n] \to [y_0, y_n]$ that approximates f and interpolates these points. The main concerns here are smoothness (continuous first or higher order derivatives) and invertibility ($f^{-1}(f(x)) = x$ for $x \in [x_0, x_n]$ and $\hat{f}_{D^{-1}} = \hat{f}_D^{-1}$) of spline function along with minimum error and gain variation. Here D^{-1} means the inverse data points (y_k, x_k). $\hat{f_D}$ has to be invertible because we want to model the nonlinear distortion in order to correct it. Furthermore $\hat{f}_{D^{-1}} = \hat{f}_D^{-1}$ ensures that we get the same model for nonlinearity regardless of which of x and y is taken as independent variable.

Although interpolative techniques such as Lagrange interpolative polynomial and cubic splines have continuous derivatives of higher order (see, e.g., [12]), they are not invertible and hence do not solve the purpose of sensor linearization. Another choice, the piecewise linear fit, is invertible, but it is not smooth and has large gain variations. It has been shown in [7] that splines of the form $x \mapsto (ax+b)/(cx+d)$ are smooth and invertible. This paper builds on the work shown in [7] and proposes a new method to find the coefficients of the spline minimizing twistiness of the interpolant and gain variations in the corrected function.

II. SPLINE CONSTRUCTION

A. Möbius Transformations

Möbius transformations are of the form $x \mapsto (ax+b)/(cx+d)$ and are associated with two-by-two matrices. As stated in [7]:

$$\left(x \mapsto \frac{ax+b}{cx+d}\right) \longleftrightarrow \left\{k\begin{bmatrix} a & b \\ c & d\end{bmatrix} : k \in \mathbf{R}\setminus 0\right\}$$

The authors are with the Center for Self-Organizing and Intelligent Systems (CSOIS), Dept. of Electrical and Computer Engineering, 4160 Old Main Hill, Utah State University, Logan, UT 84322-4160, USA. Corresponding author: Dr. YangQuan Chen. E-mail: `yqchen@ece.usu.edu`; Tel. (435)797-0148; Fax: (435)797-3054. URL: `http://www.csois.usu.edu/people/yqchen`.

Some useful properties of Möbius transformations worth recalling here are:

Property 2.1: Möbius transformations are one-on-one and onto. The inverse of the transformation $x \mapsto (ax+b)/(cx+d)$ is given by $x \mapsto (dx-b)/(-cx+a)$ which itself is a Möbius transformation, provided $ad - bc \neq 0$.

Property 2.2: Effectively there are three coefficients that define the transformation as multiplying the coefficients by a constant does not change the transformation.

Property 2.3: Composition of two Möbius transformations is the product of their respective matrices.

B. Spline Fitting

We will use the following proven result (given in [7]) as the stepping stone of our method of interpolation.

Lemma 2.1: For real numbers $x_k < x_{k+1}$ and $y_k < y_{k+1}$, let $\delta_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$. We can define a Möbius transformation g_k that maps $[x_k, x_{k+1}]$ to $[y_k, y_{k+1}]$ with the following matrix

$$\begin{bmatrix} y_{k+1} - y_k & y_k \\ 0 & 1\end{bmatrix}\begin{bmatrix} \lambda_k & 0 \\ \lambda_k - 1 & 1\end{bmatrix}\begin{bmatrix} 1 & -x_k \\ 0 & x_{k+1} - x_k\end{bmatrix} \tag{1}$$

and obeys $g_k(x_k) = y_k$, $g_k(x_{k+1}) = y_{k+1}$, $g_k'(x_k) = \lambda_k\delta_k$. This relationship can be explained using 2.3. The rightmost matrix in (1) maps $[x_k, x_{k+1}]$ to $[0, 1]$ and the leftmost one maps $[0, 1]$ to $[y_k, y_{k+1}]$. These two mappings are linear and their composition should also be linear. But the middle matrix in (1) is a mapping that fixes 0 and 1 and has a slope λ_k at 0. As can be easily verified, λ_k decides the curvature of the mapping between $[x_k, x_{k+1}]$ and $[y_k, y_{k+1}]$. Moreover, $g_k'(x_{k+1}) = \delta_k/\lambda_k$, and g_k is monotonic on $[x_k, x_{k+1}]$ if $\lambda_k > 0$. Similarly, the inverse mapping g_k^{-1} is given by

$$\begin{bmatrix} x_{k+1} - x_k & x_k \\ 0 & 1\end{bmatrix}\begin{bmatrix} \lambda_k^{-1} & 0 \\ \lambda_k^{-1} - 1 & 1\end{bmatrix}\begin{bmatrix} 1 & -y_k \\ 0 & y_{k+1} - y_k\end{bmatrix} \tag{2}$$

We define the interpolant $\hat{f} : [x_0, x_n] \to [y_0, y_n]$ such that it is a Möbius transformation, given by (1), in each subinterval $[x_k, x_{k+1}]$. But we have to find λ_k in order to determine the spline function. For the function to be continuous, we have to match derivatives at the ends of each subinterval. We know from the lemma that $g_k'(x_{k+1}) = \delta_k/\lambda_k = \lambda_{k+1}\delta_{k+1}$. Thus we can write

$$\lambda_{k+1} = \delta_k/\delta_{k+1}\lambda_k \tag{3}$$

Let us define a sequence $(e_k)_{k=0}^{n-1}$ by

$$e_0 = 1, \qquad \frac{e_{k+1}}{e_k} = \begin{cases} \delta_k/\delta_{k+1}, & \text{if} \quad k \quad \text{is} \quad \text{even} \\ \delta_{k+1}/\delta_k, & \text{if} \quad k \quad \text{is} \quad \text{odd}\end{cases} \tag{4}$$

Thus λ_k can be expressed as

$$\lambda_k = \begin{cases} \lambda_0/e_k, & \text{if} \quad k \quad \text{is} \quad \text{even} \\ e_k/\lambda_0, & \text{if} \quad k \quad \text{is} \quad \text{odd}\end{cases} \tag{5}$$

In order to find λ_0, we define the sum of all λ_k as:

$$J = a\lambda_0 + b/\lambda_0 \tag{6}$$

where the first term is the sum of λ_ks for even k and the second term is the sum of λ_ks for odd k. It can be said that J quantifies the twistiness of the spline interpolant as a function of λ_0. Thus, in order to make the spline as smooth as possible, we seek to minimize J. J attains minimum when

$$\lambda_0 = \sqrt{b/a} \tag{7}$$

Our interpolation technique can be summarized as follows:

1) Obtain the data points $D = (x_k, y_k)_{k=0}^n$.
2) Calculate $\delta_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$.
3) Find λ_0 that gives minimum curvature using (4),(5),(6) and (7).
4) Calculate $(\lambda_k)_{k=1}^n$ using (3).
5) Calculate the value of spline function $\hat{f}$ in the interval $[x_k, x_{k+1}]$ using (1) or

$$\hat{f}(x) = \frac{(y_{k+1}\lambda_k - y_k)x + y_{k+1}\lambda_k x_k + y_k x_{k+1}}{(\lambda_k - 1)x - \lambda_k x_k + x_{k+1}}$$

The inverse is given by (2).

1-4244-0210-7/06/$20.00 ©2006 IEEE

III. Simulation Results

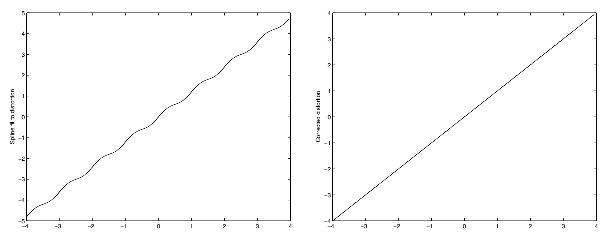
We used the distortion function as

$$f(x) = 1.2x + 0.1 \sin(2\pi x) \tag{8}$$

and sampled it at 128 equally spaced points in the interval $[-4, 4]$ to generate $D = (x_k, y_k)_{k=0}^{127}$. Using our algorithm, we found $\lambda_0 = 1.0055$ and fitted a hyperbolic spline $\hat{f}$ [Fig. 1(a)] to the data. Next we plotted the function $x \mapsto \hat{f}^{-1}(f(x))$ [Fig. 1(b)]. We observed the gain variations for the three interpolative techniques viz. linear interpolation, "minimax" method (given in [7]) and "least squares" method (one presented in this paper) [Fig. 2]. Table I compares the L_2 and L_∞ errors for the three methods. The "least squares"method clearly has an advantage over the other two methods.

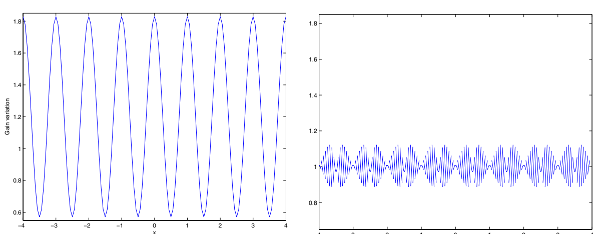
TABLE I
L_2 AND L_∞ ERRORS FOR DIFFERENT METHODS

Method	L_2 error	L_∞ error
Linear interpolation	4.04×10^{-4}	1.9×10^{-3}
Minimax	1.54×10^{-4}	5.44×10^{-4}
Least squares	4.08×10^{-5}	2.03×10^{-4}

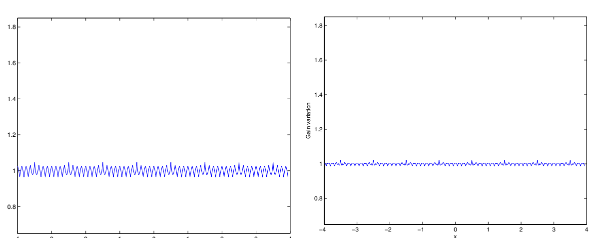


(a) Hyperbolic spline fitting before undistortion. (b) After sensor undistortion.

Fig. 1. Nonlinear distortion and compensated characteristic.



(a) After distortion (b) After correction using linear interpolation



(c) After correction using "minimax" method (d) After correction using "least squares" method

Fig. 2. Gain variations as a function of input

To investigate the effect of observation noise, we added a random value in the interval $[0, 0.05]$ to each y_k. But this would give some abrupt jumps in the data. To work around this problem we took the pointwise means for several observations of $D = (x_k, y_k)_{k=0}^{127}$. This made the grid of points smooth to some extent. We found $\lambda_0 = 0.5995$. Table II compares the L_2 and L_∞ errors for the three methods using the data infected with noise.

TABLE II
L_2 AND L_∞ ERRORS FOR DIFFERENT METHODS IN PRESENCE OF NOISE

Method	L_2 error	L_∞ error
Linear interpolation	1.3×10^{-3}	1.69×10^{-2}
Minimax	3.18×10^{-3}	4.21×10^{-2}
Least squares	2.6×10^{-3}	2.83×10^{-2}

The table shows linear interpolation to be somewhat better than the other two methods for the noisy case as linear interpolation involves less uncertainties in the calculation of parameters, i.e. slope and intercept, and hence shows robustness towards noise. But the fact that linear interpolant is not smooth prohibits its use in many applications.

IV. Conclusion

In this paper, a new method was suggested for fitting a hyperbolic spline to a set of monotonically increasing data points. The key idea of our method was to minimize the curvature of the spline in the "least-squares" sense. This method has resulted in reduced error in approximation of the distortion and less gain variations in the corrected function. Test results have shown the suitability of the fitting technique in presence of observation noise. More robust techniques have to be developed to deal with observation noise as present techniques show large gain variations in the presence of noise. Another venue of future research is to consider high order hyperbolic splines.

References

[1] K. F. Lyahou, G. van der Horn, and J. H. Huijsing, "A noniterative polynomial 2-d calibration method implemented in a microcontroller," *IEEE Trans. on Instrumentation and Measurement*, vol. 46, no. 4, pp. 752–757, August 2001.

[2] G. Picard and O. Cappe, "Blind identification of hammerstein nonlinear distortion models," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 2003, pp. 17–20.

[3] Y. Ding, H. Ohmori, and A. Sano, "Time-domain adaptive compenastion for nonlinear distortion of high power amplifiers," in *Proc. of American Control Conf.*, Denver, CO, June 2003, pp. 356–361.

[4] M. A. A. S. Choudhury, S. L. Shah, and N. F. Thornhill, "Diagnosis of poor control-loop performance using higher order statistics," *Automatica*, vol. 40, pp. 1719–1728, 2004.

[5] H. Farid, "Blind inverse gamma correction," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001.

[6] S. Maré, "Detection of nonlinear distortions in audio signals," *IEEE Trans. on Magnetics*, vol. 48, no. 2, pp. 76–80, June 2002.

[7] R. Martin, K. Lever, and J. McCarthy, "Hyperbolic splines and nonlinear distortion," *IEEE Trans. on Signal Processing*, vol. 48, no. 6, pp. 1825–1828, 2000.

[8] G. L. Dempsey, J. S. Alig, and D. E. Redfield, "Using analog neural networks for control sensor linearization," in *Proc. of the 38th Midwest Symposium on Circuits and Systems*. Rio de Janeiro, Brazil: IEEE, August 1995, pp. 73–76.

[9] N. J. Medrano-Marqués and B. M. del Brio, "Sensor linearization with neural networks," *IEEE Trans. on Industrial Electronics*, vol. 48, no. 6, pp. 1288–1290, December 2001.

[10] T. B. Bakó, T. Dabóczi, and B. A. Bell, "Automatic compensation of nonlinear distortions," in *Proc. of the IEEE Instrumentation and Measurement Technology Conf.*, Anchorage, AK, May 2002, pp. 1321–1323.

[11] K. Ooi, W. Chiang, L. Xiu, S. Tan, and Y. Chen, "Method and apparatus for determining high order polynomials for linearization of position signals," *US Patent*, no. 6,674,607, January 2004.

[12] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*, 1st ed. Boka Raton, FL: CRC Press, 1995.