# Using MACSYMA to Evaluate Likelihood Functions

Lundh, Michael

1986

*Document Version:*
Publisher's PDF, also known as Version of record

Link to publication

*Citation for published version (APA):*
Lundh, M. (1986). *Using MACSYMA to Evaluate Likelihood Functions*. (Technical Reports TFRT-7317). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
1

CODEN: LUTFD2/(TFRT-7317)/1-8/(1986)

# Using MACSYMA
# to Evaluate Likelihood Functions

Michael Lundh

Department of Automatic Control
Lund Institute of Technology
May 1986

**Department of Automatic Control**
**Lund Institute of Technology**
P.O. Box 118
S-221 00 Lund Sweden

*Document name*
Report

*Date of issue*
May 1986

*Document Number*
CODEN: LUTFD2/(TFRT-7317)/1-8/(1986)

*Author(s)*
Michael Lundh

*Supervisor*
Karl Johan Åström

*Sponsoring organisation*
STU Contract 85-4809

*Title and subtitle*
Using MACSYMA to Evaluate Likelihood Functions

*Abstract*

MACSYMA is used to evaluate the asymptotic likelihood function for a parameter estimation problem. A simple example shows how the variances of the estimated parameters are calculated in a straightforward way with some of the standard functions of MACSYMA.

*Key words*
System Identification, Formula Manipulation, Computer Algebra

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*

*ISBN*

*Language*
English

*Number of pages*
8

*Security classification*

*Recipient's notes*

## 1 INTRODUCTION

"Errare Humanum Est" is a well known fact for everyone who makes calculations by hand of huge algebraic expressions. Using MACSYMA is a way to avoid the trying hand work where it is easy to make failures. MACSYMA is an interactive program for symbolic manipulations written in LISP. Some of its facilities are derivation, matrix manipulation and equation solving. We will use MACSYMA to solve a problem that implies handling of large algebraic expressions.

## 2 THE PROBLEM

The first order system

$$y(t+1) = a_0 y(t) + b_0 u(t) + e(t+1) + c_0 e(t) \tag{1}$$

is affected by an input signal $u(t)$ and a disturbance signal $e(t)$ . The input signal is a discrete gaussian white noise process with zero mean and the variance $Eu^2(t) = \sigma^2$. The disturbance is also a discrete gaussian white noise process. It has zero mean value and unit variance $Ee^2(t) = 1$ . The two random signals are independent.
With the Maximum-Likelihood method the parameters $a_0$, $b_0$ and $c_0$ of the ARMAX model are estimated. The ML estimation method implies minimization of the loss function

$$V(a,b,c) = \frac{1}{2}\sum_{t=1}^{N} \varepsilon^2(t) \tag{2}$$

where the prediction errors $\varepsilon(t)$ are given by

$$\varepsilon(t+1) + c\varepsilon(t) = y(t+1) - ay(t) - bu(t) \tag{3}$$

We are interested in the asymptotic behaviour of the loss function $V(a,b,c)$ when $N$ approaches infinity. With this asymptotic expression the variances of the parameter estimates are calculated.

## 3 SOLUTION

The prediction error $\varepsilon(t)$ is a stochastic process. It is a sum of the outputs from two linear filters driven by the independent gaussian white noise processes $u(t)$ and $e(t)$. It can be expressed in pulse transfer operator form

$$\begin{aligned}\varepsilon(t) &= \frac{q-a}{q-c}y(t) - \frac{b}{q-c}u(t) \\ &= \frac{q^2+q(c_0-a)-c_0a}{q^2+q(c-a_0)-a_0c}e(t) + \frac{q(b_0-b)+a_0b-b_0a}{q^2+q(c-a_0)-a_0c}u(t) \\ &= H_1(q)e(t) + H_2(q)u(t)\end{aligned} \tag{4}$$

The asymptotical variance of the prediction error is given by the following integrals which are easily solved with residue-calculus.

$$E\varepsilon^2(t) = \frac{1}{2\pi i}\oint H_1(z)H_1(z^{-1})z^{-1}dzEe^2(t) + \frac{1}{2\pi i}\oint H_2(z)H_2(z^{-1})z^{-1}dzEu^2(t) \tag{5}$$

The variance of the prediction errors can be estimated when the number of measurements $N$ is sufficiently large.

$$E\varepsilon^2(t) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t) \tag{6}$$

The asymptotic expression of the loss function $V(a,b,c)$ then is

$$V(a,b,c) = \frac{N}{2}E\varepsilon^2(t) \tag{7}$$

The ML-estimate is efficient, i e. the variance of the estimated parameters are determined by the Cramer-Rao lower bound. The inverse of the Fisher-Information matrix $J$ then gives the variance of the parameter estimation.

$$E(\theta-\theta_0)(\theta-\theta_0)^T = J^{-1} \tag{8}$$

$$J = -E\frac{\partial^2(logL)}{\partial\theta^2} \tag{9}$$

It is necessary to differentiate the logarithmic Likelihood function.

$$-logL = \frac{1}{2\sigma_e^2}\sum_{t=1}^{N}\varepsilon^2(t) + Nlog\sigma_e + \frac{N}{2}log2\pi \tag{10}$$

The variance of the disturbance signal is known $(\sigma_e = 1)$ . If the constant terms of the logarithmic Likelihood function are neglected, they disappear anyway in the derivation below, the remaining term is equivalent to the loss function $V(a,b,c)$.

$$-logL = \frac{1}{2}\sum_{t=1}^{N}\varepsilon^2(t) = V(a,b,c) \tag{11}$$

Derivation of the loss function gives the Fisher-Information matrix

$$J = \begin{pmatrix} \frac{\partial^2 V}{\partial a^2} & \frac{\partial^2 V}{\partial a \partial b} & \frac{\partial^2 V}{\partial a \partial c} \\ \frac{\partial^2 V}{\partial a \partial b} & \frac{\partial^2 V}{\partial b^2} & \frac{\partial^2 V}{\partial b \partial c} \\ \frac{\partial^2 V}{\partial a \partial c} & \frac{\partial^2 V}{\partial b \partial c} & \frac{\partial^2 V}{\partial c^2} \end{pmatrix} \tag{12}$$

Evaluate this matrix at the true parameter values $a_0$, $b_0$ and $c_0$. The ML method gives unbiased estimates of the parameters of an ARMAX-process. The diagonal elements of the inverse of the matrix $J$ are expressions for the variance of the estimated parameters.

$$\begin{aligned} &E(a-a_0)^2 \\ &E(b-b_0)^2 \\ &E(c-c_0)^2 \end{aligned} \tag{13}$$

## 4 SAMPLE RUN WITH MACSYMA

The solution method described in the previous section is straightforward applied in MACSYMA.

The asymptotic variance for a stochastic process which is the output from a second order linear system driven by white noise is calculated with the function VAR2. It is found in appendix 1. The function uses the formula 6.33 in reference 1.

The macro in appendix 2 solves the problem of section 2 with the method in section 3. The results from this macro are found in appendix 3. The pulse transfer operator $H_1(q)$ is defined by the numerator vector $BB = (1 \quad c_0 - a \quad c_0 a)$ and the denominator vector $AA = (1 \quad c - a_0 \quad a_0 c)$ so that

$$H_1(q) = \frac{BB \cdot (q^2 \quad q \quad 1)^T}{AA \cdot (q^2 \quad q \quad 1)^T} \tag{14}$$

In the same way $H_2(q)$ is defined by $CC = (0 \quad b_0 - b \quad a_0 b - b_0 a)$ and $AA$. First the function VAR2 computes the integrals of (5) so that the variance of the prediction error and the loss function (2) can be formed. Derivation is a standard function in MACSYMA so forming the Fisher-Information matrix (12) is trivial. Evaluate it for the true parameter values and then invert the matrix. All this is done with simple commands. Finally extract the diagonal elements to get the variances (13). Sometimes it is necessary to simplify the expression with the commands factor and ratsimp so that the expressions not are too messy. Humans should also be able to read them, not only computers.

## 5 CONCLUDING REMARKS

Even though the system (1) is of first order hand calculation is laborious, and there are lots of occasions where errors can occur. MACSYMA is a excellent tool for this type of problem. The only difference between handling a first order system like ours or a say fourth order is the computation time. Computations like these consumes lots of processor time and it is preferable to be a single user of a VAX 11/780 when MACSYMA is used, otherwise the response times are unacceptable long.

## 6 REFERENCES


[1] Åström, K.J. and B. Wittenmark: Computer Controlled Systems. Englewood Cliffs, N.J.: Prentice Hall, Inc, 1984.

[2] Åström, K.J. (1980) Maximum Likelihood and Prediction Error Methods. Automatica 16, 551-574.

[3] Olbjer, L. Tidsserieanalys. Department of Mathematical Statistics, LTH. Lund 1985.

[4] MACSYMA Reference Manual. Cambridge Massachusetts, 1983.

```
/*-----------------------------------------------------------------
  VAR2.MAC                                               Version 02

  Function for computing intgrals for second order systems
  according to formula 6.33 in CCS.

  Inputs:  b : b-polynomial
           a : a-polynomial

  Output:  Steady state variance
  -----------------------------------------------------------------
  Author: Michael Lundh
  -----------------------------------------------------------------*/
var2(b,a):=
  Block([b0,b1,b2,a0,a1,a2,bb0,bb1,bb2,e1,num,den],

  /* Extract coeffecients in transfer function */
  b0:b[1,1],
  b1:b[1,2],
  b2:b[1,3],

  a0:a[1,1],
  a1:a[1,2],
  a2:a[1,3],

  /* Perform computations */
  bb0: b0**2 + b1**2 + b2**2,
  bb1: 2*b1*( b0 + b2 ),
  bb2: 2*b0*b2,
  e1 : a0 + a2,

  num:bb0*a0*e1 - bb1*a0*a1 + bb2*(a1*a1-a2*e1),
  den:a0*((a0*a0-a2*a2)*e1 - (a0*a1 - a1*a2)*a1),

  return:num/den)$
```

```
/* Macro for solving the problem */
writefile("Appendix3.lis");

/* Load macro for variance computation */
batchload("var2.mac")$

/* Define polynimials in transferfunctions */
bb : matrix([1,c0-a,-a*c0])$
aa : matrix([1,c-a0,-a0*c])$
cc : matrix([0,b0-b,b*a0-a*b0])$

/* Compute integrals */
V1 : var2(bb,aa)$
V2 : var2(cc,aa)$

/* Compute the loss function */
V : N/2 * ( V1 + V2*s^2 )$                /* s = sigma */
V : factor(V);

/* Differentiate the loss function */
Vaa : diff(V,a,2)$
Vab : diff(V,a,1,b,1)$
Vac : diff(V,a,1,c,1)$
Vbb : diff(V,b,2)$
Vbc : diff(V,b,1,c,1)$
Vcc : diff(V,c,2)$

/* Create Fisher Information matrix */
J : matrix([Vaa,Vab,Vac],[Vab,Vbb,Vbc],[Vac,Vbc,Vcc])$

/* Evaluate with unbiased estimate of parameters */
JJ : ev(J,a=a0,b=b0,c=c0)$
JJ : factor(JJ);

/* Invert to find Cramer-Rao lower bound */
JJI : invert(JJ)$
JJI : factor(JJI)$

/* Extract elements with variance for the estimated parameters */
Eaa : JJI[1,1];
Ebb : JJI[2,2];
Ecc : JJI[3,3];
```

(c10) V : factor(V);

(d10)
$$-\frac{n\left(\begin{aligned}&a^2 a0\, b0^2 c\, s^2 + a0\, b0^2 c\, s^2 - 2 a\, b0^2 c\, s^2 - 2 a\, a0^2 b\, b0\, c\, s^2 \\ &+ 2 a\, b\, b0\, c\, s^2 + a0^3 b^2 c\, s^2 - a0\, b^2 c\, s^2 + 2 a\, a0\, b0^2 s^2 - a^2 b0^2 s^2 \\ &- b0^2 s^2 - 2 a0^2 b\, b0\, s^2 + 2 b\, b0\, s^2 + a0^2 b^2 s^2 - b^2 s^2 + a^2 a0\, c\, c0^2 \\ &+ a0\, c\, c0^2 - 2 a\, c\, c0^2 + 2 a\, a0\, c0^2 - a^2 c0^2 - c0^2 - 2 a\, a0^2 c^2 c0 + 2 a\, c^2 c0 \\ &- 4 a\, a0\, c\, c0 + 2 a^2 c\, c0 + 2 c\, c0 + 2 a\, a0^2 c0 - 2 a^2 a0\, c0 - 2 a0\, c0 \\ &+ 2 a\, c0 + a^2 a0\, c + a0\, c - 2 a\, c + 2 a\, a0 - a^2 - 1\end{aligned}\right)}{2\,(a0 - 1)(a0 + 1)(c - 1)(c + 1)(a0\, c + 1)}$$

(c19) JJ : factor(JJ);

(d19) Col 1 =
$$\begin{bmatrix} -\dfrac{n\,(a0\, b0^2 c0\, s^2 - b0^2 s^2 + a0\, c0^3 + c0^2 - a0\, c0 - 1)}{(a0 - 1)(a0 + 1)(c0 - 1)(c0 + 1)(a0\, c0 + 1)} \\[2ex] \dfrac{b0\, c0\, n\, s^2}{(c0 - 1)(c0 + 1)(a0\, c0 + 1)} \\[2ex] \dfrac{n}{a0\, c0 + 1} \end{bmatrix}$$

Col 2 =
$$\begin{bmatrix} \dfrac{b0\, c0\, n\, s^2}{(c0 - 1)(c0 + 1)(a0\, c0 + 1)} \\[2ex] -\dfrac{n\, s^2}{(c0 - 1)(c0 + 1)} \\[2ex] 0 \end{bmatrix}$$
Col 3 =
$$\begin{bmatrix} \dfrac{n}{a0\, c0 + 1} \\[2ex] 0 \\[2ex] -\dfrac{n}{(c0 - 1)(c0 + 1)} \end{bmatrix}$$

(c22) Eaa : JJI[1,1];

(d22)
$$-\frac{(a0 - 1)(a0 + 1)(a0\, c0 + 1)^2}{n\,(b0^2 s^2 + c0^2 + 2\, a0\, c0 + a0^2)}$$

(c23) Ebb : JJI[2,2];

(d23)
$$-\frac{a0^2 b0^2 c0^2 s^2 - b0^2 s^2 + c0^4 + 2\, a0\, c0^3 + a0^2 c0^2 - c0^2 - 2\, a0\, c0 - a0^2}{n\, s^2 (b0^2 s^2 + c0^2 + 2\, a0\, c0 + a0^2)}$$

(c24) Ecc : JJI[3,3];

(d24)
$$-\frac{(c0 - 1)(c0 + 1)(b0^2 s^2 + a0^2 c0^2 + 2\, a0\, c0 + 1)}{n\,(b0^2 s^2 + c0^2 + 2\, a0\, c0 + a0^2)}$$