

Population Stochastic Modelling (PSM): Comparison with CTSM

Stig Mortensen and Søren Klim

June 7, 2008

Package: PSM, version 0.7
URL: <http://www.imm.dtu.dk/psm>

Contents

1 Purpose	1
2 Setting up the example	2
2.1 Loading data	2
2.2 Model specification	2
3 Results of comparison	4
3.1 Likelihood function value	4
3.2 Uncertainty for parameter estimates	4
3.3 State estimates	7
References	7

1 Purpose

The first stage model of PSM is based on the stochastic state space model defined in CTSM. For 'single-subject' data in a model without random effects the two programs should therefore give equal state estimates and likelihood function.

This document compares the estimation results of PSM and CTSM[1] for a fed-batch bioreactor example. The model is specified in the CTSM User's Guide[2] on page 55. The comparison is based on simulated data created in CTSM for a single bioreactor.

2 Setting up the example

The non-linear model contains three states X , S and V which are the biomass concentration, the substrate concentration and the volume. It has one input F which is the feed flow rate. There are three output/response variable $y1$, $y2$ and $y3$ which directly observes the tree states.

2.1 Loading data

Initially the data is loaded from the original file `sde0_1.csv` created by CTSM. The data file is contained in the examples folder in CTSM.

```
> library(PSM)
> tmpData <- read.table("sde0_1.csv", sep = ";",
+   col.names = c("Time", "F", "y1", "y2", "y3"))
> Time = tmpData$Time
> Y = t(tmpData[, c("y1", "y2", "y3")])
> U = t(as.matrix(tmpData[, c("F")]))
> Data <- list(Time = Time, Y = Y, U = U)
```

2.2 Model specification

The model object is specified in R as follows.

```

> MyModel <- list()
> MyModel$X0 = function(Time, phi, U) {
+   matrix(c(phi$x0, phi$s0, phi$v0), ncol = 1)
+ }
> MyModel$h = function(eta, theta, covar) {
+   phi = theta
+ }
> MyModel$Functions$f = function(x, u, time, phi) {
+   X <- x[1]
+   S <- x[2]
+   V <- x[3]
+   F <- u[1]
+   mu <- phi$mumax * S/(phi$k2 * S^2 + S + phi$k1)
+   matrix(c(mu * X - F * X/V, -mu * X/phi$Y +
+     F * (phi$sf - S)/V, F), ncol = 1)
+ }
> MyModel$Functions$df = function(x, u, time, phi) {
+   X <- x[1]
+   S <- x[2]
+   V <- x[3]
+   F <- u[1]
+   kssk = (phi$k2 * S^2 + S + phi$k1)
+   matrix(c(phi$mumax * S/kssk - F/V, phi$mumax/kssk *
+     X - phi$mumax * S/(kssk)^2 * X * (2 *
+     phi$k2 * S + 1), F * X/V^2, -phi$mumax *
+     S/(kssk)/phi$Y, -phi$mumax/(kssk) * X/phi$Y +
+     phi$mumax * S/(kssk)^2 * X/phi$Y * (2 *
+     phi$k2 * S + 1) - F/V, -F * (phi$sf -
+     S)/V^2, 0, 0, 0), nrow = 3, ncol = 3,
+     byrow = TRUE)
+ }
> MyModel$Functions$g = function(x, u, time, phi) {
+   x
+ }
> MyModel$Functions$dg = function(x, u, time, phi) {
+   diag(3)
+ }
> MyModel$S = function(u, time, phi) {
+   diag(c(phi$s11, phi$s22, phi$s33))
+ }
> MyModel$SIG = function(u, time, phi) {
+   diag(c(phi$sig11, phi$sig22, phi$sig33))
+ }

```

3 Results of comparison

3.1 Likelihood function value

The objective functions of PSM and CTSM are both the negative log-likelihood function. The two are compared based ML-estimated parameters found by CTSM. These parameter estimates are copied from CTSM and specified below. At the same time the model is prepared to estimate 5 of these parameters (**mumax**, **k1**, **s11**, **s22** and **s33**) later in this comparison.

```
> ctsmTHETA <- c(mumax = 1.0022, k1 = 0.031629,
+   s11 = 0.0075248, s22 = 0.0010636, s33 = 0.011388)
> MyModel$ModelPar = function(THETA) {
+   list(theta = list(mumax = THETA["mumax"],
+     k1 = THETA["k1"], k2 = 0.5, Y = 0.5, sf = 10,
+     sig11 = 9.6704e-28, sig22 = 1.7471e-06,
+     sig33 = 1.0903e-08, s11 = THETA["s11"],
+     s22 = THETA["s22"], s33 = THETA["s33"],
+     x0 = 1.0095, s0 = 0.23835, v0 = 1.004))
+ }
> theta <- MyModel$ModelPar(ctsmTHETA)$theta
```

The value of the CTSM negative log-likelihood function based on the parameters above are

$$\text{CTSM: } -\log L(\theta) = -388.4857 . \quad (1)$$

The negative log-likelihood function in PSM can be evaluated by directly calling the Extended Kalman Filter using the same parameter values

```
> (negLogL <- ExtKalmanFilter(theta, MyModel, Data))
[1] -388.4689
```

and the result can be seen to be very similar to the value found by CTSM as shown in Equation (1).

3.2 Uncertainty for parameter estimates

In CTSM parameter uncertainty is given by standard deviation (SD) estimates for each parameter. In PSM parameter uncertainty is given by either SD or a 95% Wald confidence interval based on the Hessian of the log-likelihood function.

To find the confidence intervals in PSM a parameter search is started with an initial guess 10% away from the CTSM ML-estimates.

```
> MyPar <- list(LB = 0.5 * ctsmTHETA, Init = ctsmTHETA *
+   1.1, UB = 1.5 * ctsmTHETA)
```

The parameter estimation is performed using the function `PSM.estimate` which returns the negative log-likelihood value, ML-estimates and the Wald confidence intervals. The result from `PSM.estimate` is shown below.

```
> (fit <- PSM.estimate(MyModel, list(Data), MyPar,
+   CI = TRUE)[1:5])

$NegLogL
[1] -388.5

$THETA
      mumax      k1      s11      s22      s33
1.002231 0.031629 0.007530 0.001064 0.011429

$CI
      mumax      k1      s11      s22      s33
Lower CI95 0.9937 0.02876 0.005452 0.0007663 0.008277
MLE        1.0022 0.03163 0.007530 0.0010640 0.011429
Upper CI95 1.0108 0.03450 0.009608 0.0013617 0.014581

$SD
      mumax      k1      s11      s22      s33
[1,] 0.004377 0.001465 0.001060 0.0001519 0.001608

$COR
      mumax      k1      s11      s22      s33
mumax 1.000e+00 9.975e-01 3.391e-02 1.680e-01 -1.844e-10
k1     9.975e-01 1.000e+00 3.388e-02 1.675e-01 -1.874e-10
s11    3.391e-02 3.388e-02 1.000e+00 5.697e-03 -8.954e-11
s22    1.680e-01 1.675e-01 5.697e-03 1.000e-00 2.589e-10
s33    -1.844e-10 -1.874e-10 -8.954e-11 2.589e-10 1.000e-00
```

The corresponding ML estimates and SD estimates in CTSM are copied from the CTSM output and shown below. The estimates in PSM and CTSM can be seen to be almost equal.

```
> ctsmEstimates

      mumax      k1      s11      s22      s33
MLE 1.002200 0.031629 0.0075248 0.0010636 0.011388
SD  0.003342 0.001598 0.0009762 0.0001376 0.001496
```

To visually compare the error estimates in PSM and CTSM a Wald confidence interval is generated from CTSM as $MLE \pm 1.96 \cdot SD$. This result of the comparison is shown in Figure 1.

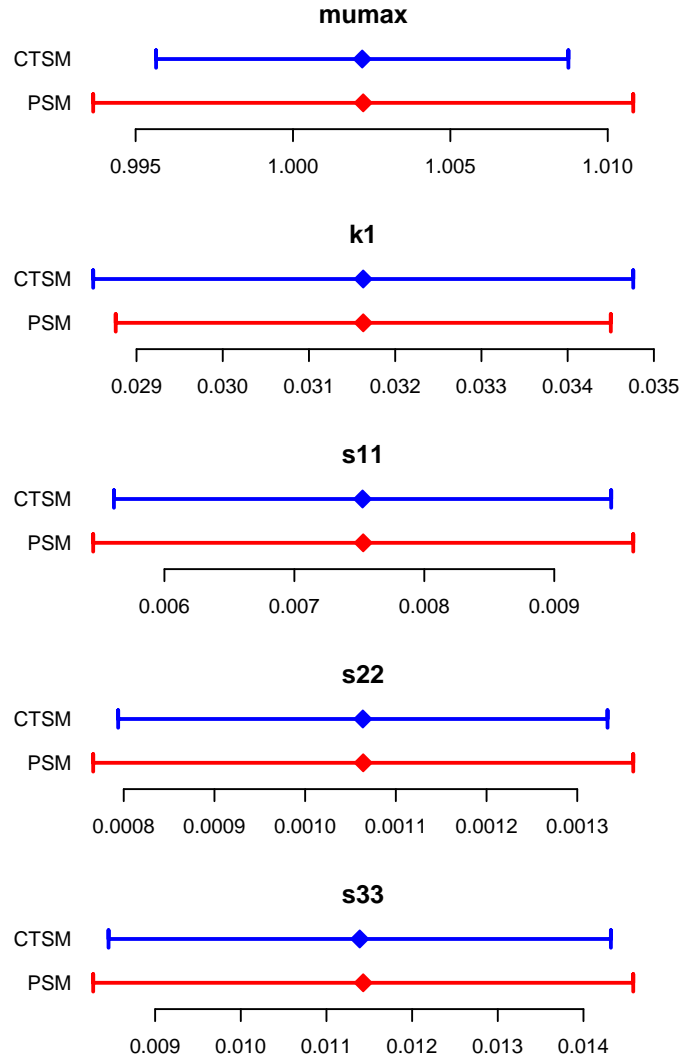


Figure 1: Visual comparison of error estimates for ML-estimated parameters in PSM and CTSM.

3.3 State estimates

The state estimates in PSM and CTSM should also be equal. This is expected to be the case for the state and state covariance one-step predictions since the likelihood function is evaluated based on these.

The smoothed state estimate is the optimal state estimate where all data is used at each time point. To compare these a smoothed state estimate is generated in CTSM and loaded into R in the variable `CTSMSmooth`. The two first rows of this variable are shown below.

```
> CTSMSmooth[1:2, ]
```

	Time	Xs1	Xs2	Xs3	SD.Xs1	SD.Xs2	SD.Xs3
1	0.000	1.010	0.2383	1.004	6.794e-11	3.391e-07	2.125e-09
2	0.038	1.032	0.2545	1.012	2.342e-10	4.774e-07	3.006e-09

The state estimates are then created in PSM using the command shown below.

```
> sm1 <- PSM.smooth(MyModel, list(Data), ctsmTHETA)[[1]]
```

The state estimates from PSM and CTSM are compared in Figure 2. Based on a visual inspection these can be seen to be very alike.

An exact numerical analysis of the differences can also be performed as shown in the line below.

```
> (pct <- max(abs(t(CTSMSmooth[, 2:4]) - sm1$Xs)/sm1$Xs *  
+      100))  
[1] 0.07457
```

As it can be seen the analysis shows that smoothed state estimates differ at most 0.075%.

References

- [1] Kristensen NR, Madsen H (2003) Continuous time stochastic modelling: CTSM 2.3 mathematics guide, Technical University of Denmark
<http://www2.imm.dtu.dk/ctsm/MathGuide.pdf>
- [2] Kristensen NR, Madsen H (2003) Continuous time stochastic modelling: CTSM 2.3 User's Guide, Technical University of Denmark
<http://www2.imm.dtu.dk/ctsm/UsersGuide.pdf>

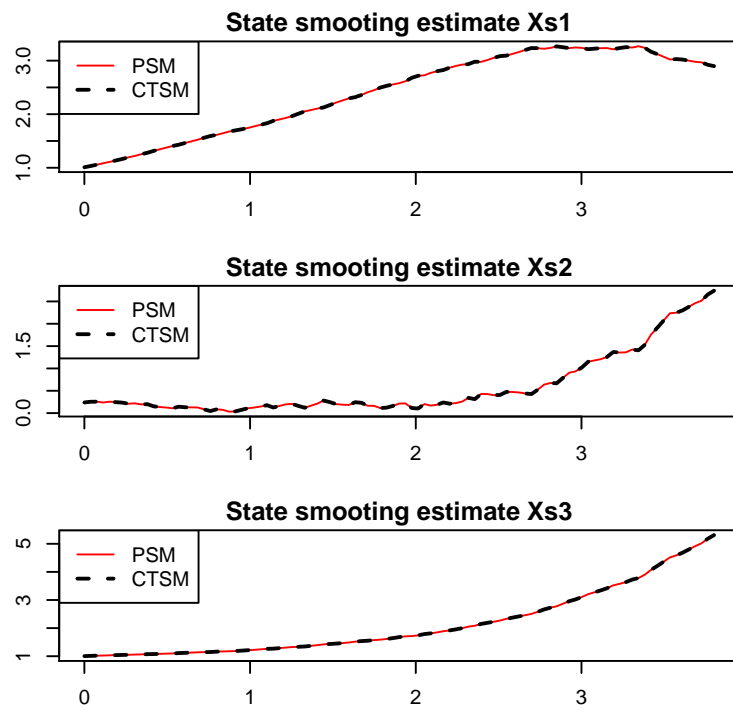


Figure 2: Comparison of smoothed estimate of states.