

Generating the results of Table 2 involves a number of Eviews programs, displayed below.

The raw simulation data is generated by `com_statics04.prg`. In this program for each  $0, z, h$  (where  $z = h/3$ ) a separate working file is created. I actually ran this a separate batches on a number of machines. Then program `consolidate04.prg` merges all this simulated data into one data file (eviews working file).

Program `define1.prg` defines each of the non-linear models and estimates them over all the simulated data. In some cases finding good starting values involved some tricks like first estimating a simplified version of the model. The first time it was run commands like `c=smooth1_c` were commented out. Thereafter the "param" commands were commented out, and comments like "`c=smooth1_c`" were turned back to commands (comment markers removed). Likewise any simplified versions were commented out. So then the program could be run multiple times, each of them updating the coefficient vector saved, (such as `smooth1_c`).

Then programs `big_forecastA.prg` through `big_forecastD.prg` estimates the models in the two subsamples, saving the resulting coefficient vectors for use in the next program.

Finally, program `big_forecast.prg` uses coefficients as estimated in one part of the sample to forecast in the other part. It then creates a table called "forecast1" in which forecasting and re-estimation results are saved, which are displayed in Table 2 of the paper.

program com\_statics04  
Eviews program com\_statics04.prg

```
""program com_statics04
""note standard Miller-Orr with NORMAL net receipts. Will allow fractional Z,H
""Note this creates separate workfiles for each z, so not using sss
""Static variation in b
""Aug 11, 2006

""WFCreat(wf=com_static04) u 20
rndseed(default = "mt4")
!neach = 100000
For %week 10
!sets = 0
!bcount = 1
For %z 11.2 13.7
""3.61 3.86 4.23 4.61 4.98 5.41 5.95 6.44 7.24 8.2 9.3 11.2 13.7 20.5
WFCreat(wf=com_static04_{%z}) u 100000
smpl 1 10000
!bigstart = !sets*!neach + 1 ""will be recording !neach obs for every Z
!sets = !sets + 1
matrix(10000,2) allM
!z = {%z}
!H = 3*!z
!PZ = 2/(!H)
!slope1 = !PZ/(!z)
!slope2 = !PZ/(2*!z)
genr set1 = !z*(4/3) ""at mean if not otherwise modified
genr set2 = !z*(4/3)

!smplstart = 0
!smplend = 0
For !x = !z to 0 step -1 ""initialize to just above zero
!smplstart = !smplend + 1
!smplend = !smplstart + @floor( (!x)*!slope1*10000 ) - 1
If !smplstart <= !smplend Then
smpl !smplstart !smplend
genr set1 = !x
genr set2 = !x
Endif
Next
""still using above !smplstart and end

For !x = !z+1 to !H ""initialize through H
!smplstart = !smplend + 1
!smplend = !smplstart + @floor( (!PZ - (!x-!z)*!slope2)*10000 ) - 1
If !smplstart <= !smplend Then
smpl !smplstart !smplend
genr set1 = !x
genr set2 = !x
Endif
Next

smpl 1 10000
""now begin to shuffle, pair, assign net, record number of adjustments andadjust to targets, sum
""will want to see if number of adjustments correlated with sum being high or low.
For !ini = 1 to 1000 ""simulate a bunch of rounds to initialize
stom(set1,vec1)
stom(set2,vec2) ""converted series to vectors
colplace(allM,vec1,1) ""place vectors in a matrix
colplace(allM,vec2,2)
```



## program com\_statics04

```
smpl if set1 <=0  ""reallocate if breach interval
!trigger0 = !trigger0 + @obs(set1)
genr set1 = !z +set1
smpl 1 10000
smpl if set1 <=0  ""reallocate if breach interval 2nd time
genr set1 = !z +set1
smpl 1 10000

smpl if set1 >= !H
!triggerH = !triggerH + @obs(set1)
genr set1 = !z +(set1 -!H)
smpl 1 10000
smpl if set1 >= !H          ""2ndtime
genr set1 = !z +(set1 -!H)
smpl 1 10000

smpl if set2 <=0  ""reallocate if breach interval
!trigger0 = !trigger0 + @obs(set2)
genr set2 = !z +set2
smpl 1 10000
smpl if set2 <=0  ""reallocate if breach interval 2nd time
genr set2 = !z +set2
smpl 1 10000

smpl if set2 >= !H
!triggerH = !triggerH + @obs(set2)
genr set2 = !z +(set2 -!H)
smpl 1 10000
smpl if set2 >= !H          ""2nd time
genr set2 = !z +(set2 -!H)
smpl 1 10000

!sumn = 0
Next "" For !week = 1 to
""!sss = !bigstart +!round -1  only use if making one big workfile
""smpl !sss !sss
smpl !round !round
genr porTrigger = (!trigger0 +!triggerH)/20000
genr porTrigger0 = !trigger0/20000
genr porTriggerH = !triggerH/20000
genr Z = %Z
!trigger0 = 0
!triggerH = 0
smpl 1 10000
!sumn = @sum(set1) +@sum(set2)
""smpl !sss !sss
smpl !round !round
genr total = !sumn
smpl 1 10000
Next ""For !round= 1 to

""smpl !bigstart !bigstart+!neach-1  ""this was for one big file
smpl 1 10000
!s total = c(1)
genr av = c(1)
""!av = c(1)
""!s total = c(2)*total(-1) +(1 -c(2))*av
""!b2 = c(2)
smpl 1 10000
WFsave com_static04_{%z}.wfl
```

```
program com_statics04
close com_static04_{%z}.wfl
""WFselect com_static04
""smp1 !bcount !bcount
""genr av = !av
""genr bM = !b2
""genr z = %z
!bcount = !bcount +1
Next ""For %z

Next ""For %week 10
""WFsave com_static04
""smp1 1 14
""param 1 -10 2 0
""ls bm = exp(c(1)/(c(2)+z^2))
```

program consolidate04  
Eviews program consolidate04.prg

```
""program consolidate04
""Sept 5, 2006
""consolidates individual data from programs com_statics04
""but uses only 10,001 obs from each file (each z)
!file = 1
For %z 3.61 3.86 4.23 4.61 4.98 5.41 5.95 6.44 7.24 8.20 9.30 11.2 13.7 20.5
  WFopen com_static04_{%z}.wf1
  smpl 2 10001
  genr m_{!file} = total
  genr mlag_{!file} = total(-1)
  genr av_{!file} = av
  genr z_{!file} = z
  store(i) m_{!file} mlag_{!file} av_{!file} z_{!file}
  close com_static04_{%z}.wf1
  !file = !file +1
Next
""above stores obs of individual series

WFCreate(wf=com_static04_c) u 140030
!start = 0
!round = 1
For !file = 1 to 14
  fetch m_{!file} mlag_{!file} av_{!file} z_{!file} ""these are complete for smpl 2 10001
  !sstart = (!start*10000) +1
  !leend = !sstart +9999
  !lag = ((!round-1)*10000) -1
  smpl !sstart !leend
  genr M = m_{!file}(-!lag)
  genr M_lag = mlag_{!file}(-!lag)
  genr av = av_{!file}(-!lag)
  genr z = z_{!file}(-!lag)
  !start = !start +1
!round = !round +1
Next
```

program define1.prg  
Eviews program define1.prg

""Program define1.prg

""Defines and estimates non-linear models

smpl 1 140000

""c(1) = -1

""c = MBW\_c

""equation MBW.ls(m=1000,c=0.0000001,s) m = (EXP((10^10)\*C(1)/(AV^2)))\*M\_LAG + (1 - (EXP((10^10)\*C(1)/(AV^2))))\*AV

""coef(751) MBW\_c = c

""""Define smooth-exponential

""param 1 0.1 3 0.001

""ls m = (C(1) + 1\*(EXP((-C(3)\*(((10^-5)\*(M\_LAG - AV) - 0\*(10^5))^2)))))\*M\_LAG + (1 - (C(1) + 1\*(EXP((-C(3)\*(((10^-5)\*(M\_LAG - AV) - 0\*(10^5))^2))))))\*AV

""param 2 1 4 0

""c = smooth1\_c

""equation smooth1.ls(m=1000,c=0.0000001,s) m = (C(1) + C(2)\*(EXP((-C(3)\*(((10^-5)\*(AV - M\_LAG) - C(4)\*(10^5))^2)))))\*M\_LAG + (1 - (C(1) + C(2)\*(EXP((-C(3)\*(((10^-5)\*(AV - M\_LAG) - C(4)\*(10^5))^2))))))\*AV

""coef(751) smooth1\_c = c

""genr ser1 = smooth1.@se

""c = 0

""param 2 1 3 0.01

""ls m = (0 + c(2)\*(EXP((-C(3)\*(((10^-5)\*(M\_LAG - AV) - 0\*(10^5))^2)))))\*M\_LAG + (1 - (0 + c(2)\*(EXP((-C(3)\*(((10^-5)\*(M\_LAG - AV) - 0\*(10^5))^2))))))\*AV

""smooth1.ls(m=2000,c=0.0000001,s)

""coef(751) smooth1\_c2 = c

""genr ser2 = smooth1.@se

""""""restricted smooth-exponential

""param 1 0 2 0.01 3 0.01 4 0

""ls(m=100,c=0.0000001,s) m = (0 + (C(2)^2)\*(EXP((-C(3)^2)\*(((10^-5)\*(AV - M\_LAG) - 0\*(10^5))^2)))))\*M\_LAG + (1 - (0 + (C(2)^2)\*(EXP((-C(3)^2)\*(((10^-5)\*(AV - M\_LAG) - 0\*(10^5))^2))))))\*AV

c = smooth1r\_c

equation smooth1r.ls(m=1000,c=0.0000001,s) m = (C(1) + (C(2)^2)\*(EXP((-C(3)^2)\*(((10^-5)\*(AV - M\_LAG) - C(4)\*(10^5))^2)))))\*M\_LAG + (1 - (C(1) + (C(2)^2)\*(EXP((-C(3)^2)\*(((10^-5)\*(AV - M\_LAG) - C(4)\*(10^5))^2))))))\*AV

coef(751) smooth1r\_c = c

store(i) smooth1r

store(i) smooth1r\_c

""""""Define smooth-logistic

param 1 1 2 0.01 3 0.01 4 0

ls(m=200,c=0.0000001,s) m = (1 - (C(2))/(1 + EXP(-C(3)\*(10^-5)\*(M\_LAG - AV + 0)))))\*M\_LAG + (1 - (1 - (C(2))/(1 + EXP(-C(3)\*(10^-5)\*(M\_LAG - AV + 0))))))\*AV

""c = smooth2\_c

Equation smooth2.ls(m=1000,c=0.0000001,s) m = (c(1) - (C(2))/(1 + EXP(-C(3)\*(10^-5)\*(M\_LAG - AV + c(4))))))\*M\_LAG + (1 - (c(1) - (C(2))/(1 + EXP(-C(3)\*(10^-5)\*(M\_LAG - AV + c(4))))))\*AV

coef(751) smooth2\_c = c

store(i) smooth2

store(i) smooth2\_c

""""""restricted smooth-logistic

param 1 1 2 0.01 3 0.01 4 0

ls(m=200,c=0.0000001,s) m = (1 - ((c(2)^2))/(1 + EXP(-((c(3)^2)\*(10^-5)\*(M\_LAG - AV + 0))))))\*M\_LAG + (1 - (1 - ((c(2)^2))/(1 + EXP(-((c(3)^2)\*(10^-5)\*(M\_LAG - AV + 0))))))\*AV

program define1.prg

Equation smooth2r.ls(m=1000,c=0.0000001,s) m = ( C(1) -((c(2)^2))/(1 +EXP( -(c(3)^2))\*(10^-5)\*(M\_LAG -AV +C(4)))  
))\*M\_LAG + (1 - ( C(1) -((c(2)^2))/(1 +EXP( -(c(3)^2))\*(10^-5)\*(M\_LAG -AV +C(4))) )) ) \*AV

coef(751) smooth2r\_c = c

store(i) smooth2r

store(i) smooth2r\_c

programs big\_forecastA.prg, big\_forecastB, big\_forecastC, big\_forecastD  
Eviews programs big\_forecastA-D:

```
""program big_forecastA
""workfile com_static04_c c is for consolidated
""uses first half of smpl to forecast in second half
""BREAKS UP AND SAVES STARTING COEFS FOR SMOOTH1
""March 23, 2007
```

```
smpl 1 70000
c = smooth1_c
smooth1.ls(m=2000,c=0.0000001,s)
coef(751) smooth1F_c = c ""F for first
smpl 70001 140000
c = smooth1_c
smooth1.ls(m=2000,c=0.0000001,s)
coef(751) smooth1S_c = c """" S for second
```

```
store(i) smooth1F_c
store(i) smooth1S_c
```

```
-----
""program big_forecastB
""workfile com_static04_c c is for consolidated
""uses first half of smpl to forecast in second half
""BREAKS UP AND SAVES STARTING COEFS FOR SMOOTH1R
""March 23, 2007
```

```
smpl 1 70000
c = smooth1r_c
smooth1r.ls(m=2000,c=0.0000001,s)
coef(751) smooth1rF_c = c ""F for first
smpl 70001 140000
c = smooth1r_c
smooth1r.ls(m=2000,c=0.0000001,s)
coef(751) smooth1rS_c = c """" S for second
```

```
store(i) smooth1rF_c
store(i) smooth1rS_c
```

```
-----
""program big_forecastC
""workfile com_static04_c c is for consolidated
""uses first half of smpl to forecast in second half
""BREAKS UP AND SAVES STARTING COEFS FOR SMOOTH2
""March 23, 2007
```

```
smpl 1 70000
c = smooth2_c
smooth2.ls(m=2000,c=0.0000001,s)
coef(751) smooth2F_c = c ""F for first
smpl 70001 140000
c = smooth2_c
smooth2.ls(m=2000,c=0.0000001,s)
coef(751) smooth2S_c = c """" S for second
```

```
store(i) smooth2F_c
store(i) smooth2S_c
```

programs big\_forecastA.prg, big\_forecastB, big\_forecastC, big\_forecastD

-----

```
"""program big_forecastD
"""workfile com_static04_c c is for consolidated
"""uses first half of smpl to forecast in second half
"""BREAKS UP AND SAVES STARTING COEFS FOR smooth2rr
"""March 23, 2007
```

```
smpl 1 70000
c = smooth2r_c
smooth2r.ls(m=2000,c=0.0000001,s)
coef(751) smooth2rF_c = c  """F for first
smpl 70001 140000
c = smooth2r_c
smooth2r.ls(m=2000,c=0.0000001,s)
coef(751) smooth2rS_c = c  """S for second
```

```
store(i) smooth2rF_c
store(i) smooth2rS_c
```

program big\_forecast  
Eviews program big\_forecast.prg

```
""program big_forecast  
""workfile com_static04_c c is for consolidated  
""FIRST RUN PROGRAM DEFINE1  
""THEN RAN PROGRAMS big_forecastA, big_forecastB, ...C ...D which saved near convergence starting values  
""which were fetched into the working file.  
""uses first half of smpl to forecast in second half  
""March 23, 2007
```

```
smpl 1 70000  
equation Lin.ls m = c(1)*m_lag +c(2)*av  
smpl 70001 140000  
gener mhat_Lin = c(1)*m_lag +c(2)*av  
Lin.LS  
!ser_Lin = Lin.@se
```

```
smpl 1 70000  
c(1) = -1  
equation MBW.ls(m=2000,c=0.0000001,s) m = (EXP((10^10)*C(1)/(AV^2)))*M_LAG + (1 -(EXP((10^10)*C(1)/(AV^2))))*AV  
smpl 70001 140000  
gener mhat_MBW = (EXP((10^10)*C(1)/(AV^2)))*M_LAG + (1 -(EXP((10^10)*C(1)/(AV^2))))*AV  
c(1) = -1  
MBW.ls(m=2000,c=0.0000001,s)  
!ser_MBW = MBW.@se
```

```
smpl 1 70000  
c = smooth1F_c  
smooth1.ls(m=20,c=0.0000001,s)  
smpl 70001 140000  
gener mhat_smooth1 = (C(1) +C(2)*(EXP((-C(3)*(((10^-5)*(AV -M_LAG) -C(4)*(10^5) )^2) ))) ) *M_LAG + (1- (C(1)  
+C(2)*(EXP((-C(3)*(((10^-5)*(AV -M_LAG) -C(4)*(10^5) )^2) ))) ) ) *AV  
c = smooth1S_c  
smooth1.ls(m=20,c=0.0000001,s)  
!ser_smooth1 = smooth1.@se
```

```
smpl 1 70000  
c = smooth1rF_c  
smooth1r.ls(m=20,c=0.0000001,s)  
smpl 70001 140000  
gener mhat_smooth1r = (C(1) +(C(2)^2)*(EXP((-C(3)^2)*(((10^-5)*(AV -M_LAG) -C(4)*(10^5) )^2) ))) *M_LAG + (1- (C(1)  
+(C(2)^2)*(EXP((-C(3)^2)*(((10^-5)*(AV -M_LAG) -C(4)*(10^5) )^2) ))) ) *AV  
c = smooth1rS_c  
smooth1r.ls(m=20,c=0.0000001,s)  
!ser_smooth1r = smooth1r.@se
```

```
smpl 1 70000  
c = smooth2F_c  
smooth2.ls(m=20,c=0.0000001,s)  
smpl 70001 140000  
gener mhat_smooth2 = ( c(1) -(C(2))/(1 +EXP( -(C(3))*(10^-5)*(M_LAG -AV +c(4))) ) ) *M_LAG + (1- (c(1) -(C(2))/(1 +EXP( -  
(C(3))*(10^-5)*(M_LAG -AV +c(4))) ) ) ) *AV  
c = smooth2S_c  
smooth2.ls(m=20,c=0.0000001,s)  
!ser_smooth2 = smooth2.@se
```

```
smpl 1 70000  
c = smooth2rF_c  
smooth2r.ls(m=20,c=0.0000001,s)  
smpl 70001 140000
```

```

program big_forecast
genr mhat_smooth2r = ( C(1) - ((c(2)^2))/(1 + EXP( -((c(3)^2))*(10^-5)*(M_LAG -AV +C(4)))) ))*M_LAG + (1 - ( C(1) -
((c(2)^2))/(1 + EXP( -((c(3)^2))*(10^-5)*(M_LAG -AV +C(4)))) ) ))*AV
c = smooth2rF_c
smooth2r.ls(m=20,c=0.0000001,s)
!ser_smooth2r = smooth2r.@se

table(7,3) forecast1
forecast1(1,1) = "model"
forecast1(1,2) = "rmse2ndhalf"
forecast1(1,3) = "SER 2ndhalf"

forecast1(2,1) = "Linear"
forecast1(3,1) = "Smooth1"
forecast1(4,1) = "Smooth1r"
forecast1(5,1) = "Smooth2"
forecast1(6,1) = "Smooth2r"
forecast1(7,1) = "Exp01"

smp1 70001 140000
genr err = m -mhat_Lin
genr err2 = err^2
!merr2 = @sum(err2)/@obs(err2)
!rmse = !merr2^0.5
forecast1(2,2) = !rmse
forecast1(2,3) = !ser_Lin
d err err2

genr err = m -mhat_smooth1
genr err2 = err^2
!merr2 = @sum(err2)/@obs(err2)
!rmse = !merr2^0.5
forecast1(3,2) = !rmse
forecast1(3,3) = !ser_smooth1
d err err2

genr err = m -mhat_smooth1r
genr err2 = err^2
!merr2 = @sum(err2)/@obs(err2)
!rmse = !merr2^0.5
forecast1(4,2) = !rmse
forecast1(4,3) = !ser_smooth1r
d err err2

"""
genr err = m -mhat_smooth2
genr err2 = err^2
!merr2 = @sum(err2)/@obs(err2)
!rmse = !merr2^0.5
forecast1(5,2) = !rmse
forecast1(5,3) = !ser_smooth2
d err err2
"""

genr err = m -mhat_smooth2r
genr err2 = err^2
!merr2 = @sum(err2)/@obs(err2)
!rmse = !merr2^0.5
forecast1(6,2) = !rmse
forecast1(6,3) = !ser_smooth2r
d err err2

```

```
program big_forecast
```

```
genr err = m -mhat_MBW  
genr err2 = err^2  
!merr2 = @sum(err2)/@obs(err2)  
!rmse = !merr2^0.5  
forecast1(7,2) = !rmse  
forecast1(7,3) = !ser_MBW  
d err err2
```