# Reg huSEM Appendix: huSEM to VAR

In this code we generate a huSEM and transform it to an equivalent VAR. It is supplemental material to the Appendix in the paper entitled, "Directionality discovery in individual dynamic models: A regularized unified structural equation modeling approach for hybrid vector autoregression".

## Data generating matrices

Let's start with making the $A$,$\Phi^*$, and$\Phi$ matrices for a 3-variate model with one contemporaneous directed relation ($A_{2,1}$) and one contemporaneous bidirectional relation among residuals ($\Phi_{1,3} = \Phi_{3,1}$).

```
A <- matrix(c(0,0,0,.4,0,0,0,0,0), nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]  0.0    0    0
## [2,]  0.4    0    0
## [3,]  0.0    0    0
```

```
Phistar <- matrix(c(.5,0,0,0,.6,0,0,0,.2), nrow = 3, ncol = 3, byrow = TRUE)
Phistar
```

```
##      [,1] [,2] [,3]
## [1,]  0.5  0.0  0.0
## [2,]  0.0  0.6  0.0
## [3,]  0.0  0.0  0.2
```

```
Psi <- matrix(c(1,0,.5,0,1,0,.5,0,1), nrow = 3, ncol = 3, byrow = TRUE)
Psi
```

```
##      [,1] [,2] [,3]
## [1,]  1.0    0  0.5
## [2,]  0.0    1  0.0
## [3,]  0.5    0  1.0
```

## Simulating Data

Let's make some data!

First we need the $(I - A)^{-1}$ matrix:

```
I   <- diag(3)
neg <- solve(I-A)
```

From here we just need to insert our data generting matrices and initiate our time series vector by generating first our noise vector:

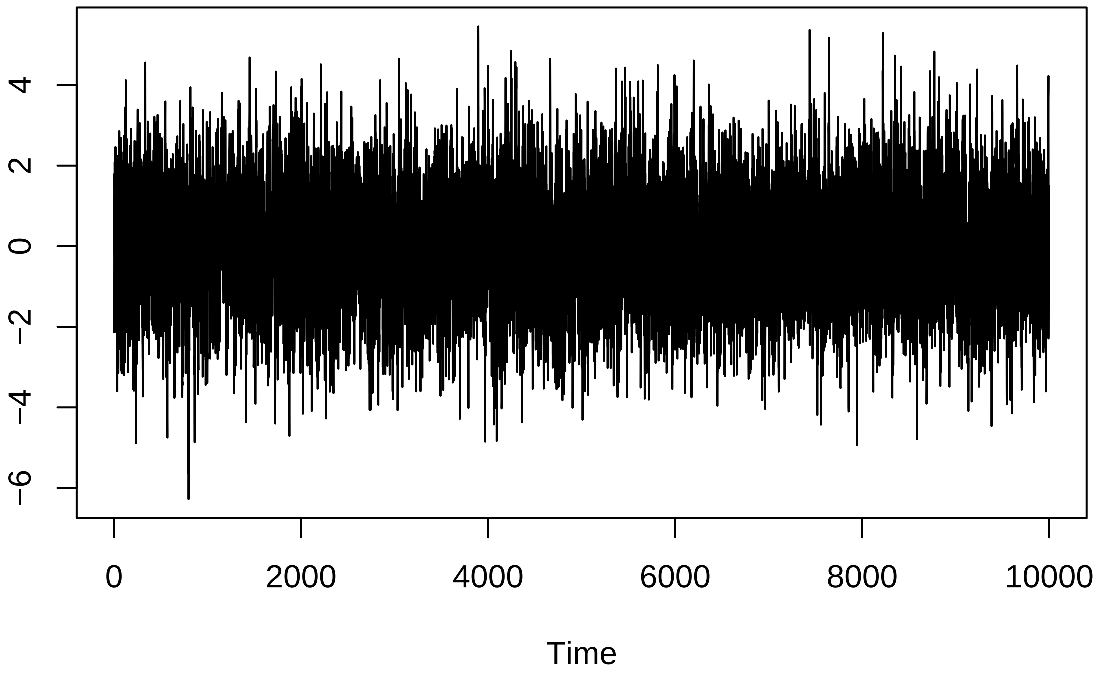
```
library(MASS)
noise <- neg %*% t(mvrnorm(n = 3*10000, rep(0,3), Psi, empirical = TRUE))
time <- time1 <- matrix(0, nrow = 3, ncol = 10000)
time[,1] <- noise[,1]                                    # Contemporaneous
time1[,1] <- neg %*% Phistar %*% time[,1] + noise[,1]    # Lag(1)
```

```
time[,2] <- time1[,1]
for (j in 2:(10000)){
  time1[,j]  <- neg %*% Phistar %*% time[,(j)] + noise[,j]
  if (j<10000)
    time[,(j+1)] <- time1[,j]
}
```

You might note that this transformation is exploited directly in the data-generating process.

Let's take a look, make sure it doesn't explode or anything:

```
ts.plot(t(time))
```



Looks good.

## VAR

Now we fit a VAR to this data.

```
library(vars)
fitVAR <- VAR(t(time),
              lag.max = 1,
              ic = c("AIC"))
coef(fitVAR)
```

```
## $y1
##             Estimate  Std. Error    t value   Pr(>|t|)
## y1.l1  0.5102455238 0.010840148 47.0699764 0.00000000
## y2.l1  0.0146131565 0.007538476  1.9384762 0.05259328
## y3.l1 -0.0107214152 0.011254131 -0.9526649 0.34078294
## const -0.0001709943 0.010071111 -0.0169787 0.98645395
##
## $y2
##            Estimate  Std. Error    t value     Pr(>|t|)
## y1.l1  0.190802713 0.011578204 16.4794747 3.197467e-60
## y2.l1  0.620341481 0.008051736 77.0444385 0.000000e+00
## y3.l1 -0.008501788 0.012020372 -0.7072816 4.794080e-01
## const  0.001927985 0.010756807  0.1792340 8.577576e-01
##
## $y3
##           Estimate  Std. Error   t value     Pr(>|t|)
## y1.l1 -0.005763343 0.010780260 -0.534620 5.929245e-01
## y2.l1  0.014329324 0.007496828  1.911385 5.598371e-02
## y3.l1  0.196758211 0.011191955 17.580325 3.678204e-68
## const -0.014062645 0.010015472 -1.404092 1.603225e-01
```

Putting those coefficients into a more familiar $\Phi$ form (rounding to the tenths spot):

$$\Phi = \begin{bmatrix} .5 & 0 & 0 \\ .2 & .6 & 0 \\ 0 & 0 & .2 \end{bmatrix}, \tag{1}$$

This is precisely what we'd expect from our transformation when done algebraically:

```
Phi = neg %*% Phistar
Phi
```

```
##      [,1] [,2] [,3]
## [1,]  0.5  0.0  0.0
## [2,]  0.2  0.6  0.0
## [3,]  0.0  0.0  0.2
```

Now we take a look at the covariance of residuals. We would expect $\Theta$ to be:

```
Theta <- neg %*% Psi %*% t(neg)
Theta
```

```
##      [,1] [,2] [,3]
## [1,]  1.0 0.40  0.5
## [2,]  0.4 1.16  0.2
## [3,]  0.5 0.20  1.0
```

What do we get in this toy example?

```
zeta <- residuals(fitVAR)
round(cov(zeta),2)
```

```
##      y1   y2   y3
## y1 1.01 0.40 0.52
## y2 0.40 1.16 0.20
## y3 0.52 0.20 1.00
```

This is nearly exactly what we'd get analytically.