Component response rate variation drives stability in large complex systems

Supporting information

Brad Duthie

Code and simulations underlying Fig. 1

The sample M used for the eigenvalue distributions in Fig. 1 of the text is available on GitHub, and was produced with by running the following function.

```
find_bgamma <- function(S = 200, C = 0.05, Osd = 0.4, iters = 10000){
    while(iters > 0){
        A_dat  <- rnorm(n = S * S, mean = 0, sd = Osd);
        A_mat  <- matrix(data = A_dat, nrow = S);
        C_dat  <- rbinom(n = S * S, size = 1, prob = C);
        C_mat  <- matrix(data = C_dat, nrow = S, ncol = S);
        A_mat  <- A_mat * C_mat;
        gammas <- c(rep(1.95, S/2), rep(0.05, S/2))
        mu_gam <- mean(gammas);
        diag(A_mat) <- -1;
        A1     <- gammas * A_mat;
        A0     <- mu_gam * A_mat;
        A0_e   <- eigen(A0)$values;
        A0_r   <- Re(A0_e);
        A0_i   <- Im(A0_e);
        A1_e   <- eigen(A1)$values;
        A1_r   <- Re(A1_e);
        A1_i   <- Im(A1_e);
        if(max(A0_r) >= 0 & max(A1_r) < 0){
            return(list(A0 = A0, A1 = A1));
            break;
        }
        print(iters);
        iters <- iters - 1;
    }
}
```

The above function terminates when a matrix M is found that is not stable when all component response rates are set to $\gamma = 1$, but is stable when half of component response rates are 1.95 and half are 0.05. The function is used to illustrate the concept of how a fast versus slow component responses can cause a system to become stable. Simulations were run for `iter = 1000000`, but terminated once an acceptable `A0` and `A1` were found. The code below plots the eigenvalue distributions of `A0` and `A1` in panels **a** and **b**, respectively.

The plot itself can be recreated with the code below.

```
par(mfrow = c(1, 2), mar = c(0.5, 0.5, 0.5, 0.5), oma = c(5, 5, 0, 0));
plot(A0_r, A0_i, xlim = c(-3.7, 0.3), ylim = c(-2, 2), pch = 4, cex = 0.7,
     xlab = "", ylab = "", cex.lab = 1.5, cex.axis = 1.5, asp = 1);
vl <- seq(from = 0, to = 2*pi, by = 0.001);
A0x0 <- sqrt(200) * sd(A0vec) * cos(vl) + mean(diag(A0));
A0y0 <- sqrt(200) * sd(A0vec) * sin(vl);
```

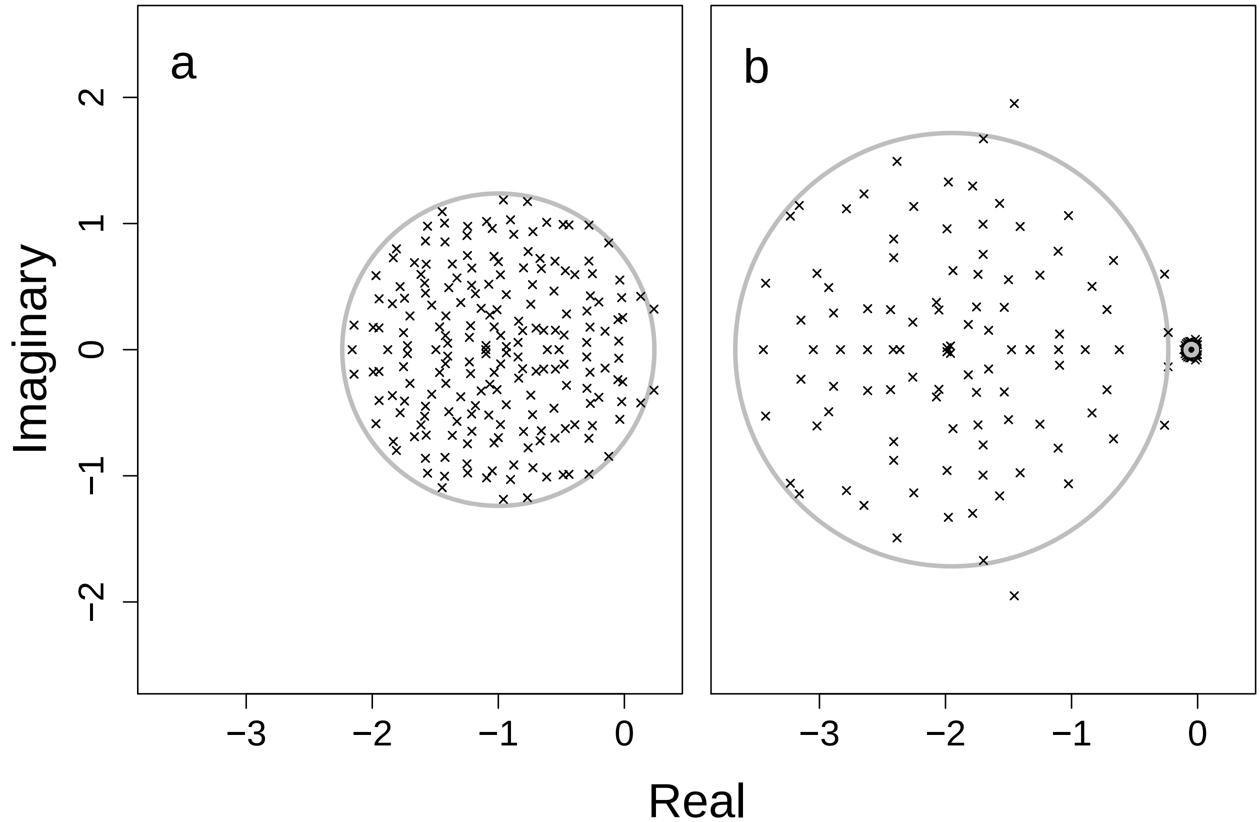
```
text(x = -3.5, y = 2.25, labels = "a", cex = 2);
points(x = A0x0, y = A0y0, type = "l", lwd = 3, col = "grey");
points(A0_r, A0_i, pch = 4, cex = 0.7);

plot(A1_r, A1_i, xlim = c(-3.7, 0.3), ylim = c(-2, 2), pch = 4, cex = 0.7,
     xlab = "", ylab = "", cex.lab = 1.5, cex.axis = 1.5, asp = 1,
     col = "black", yaxt = "n");

vl <- seq(from = 0, to = 2*pi, by = 0.001);
A0x1a <- sqrt(100) * sd(A1vec[fhalf]) * cos(vl) + mean(diag(A1)[1:100]);
A0y1a <- sqrt(100) * sd(A1vec[fhalf]) * sin(vl);
points(x = A0x1a, y = A0y1a, type = "l", lwd = 3, col = "grey");
A0x1b <- sqrt(100) * sd(A1vec[shalf]) * cos(vl) + mean(diag(A1)[101:200]);
A0y1b <- sqrt(100) * sd(A1vec[shalf]) * sin(vl);
points(x = A0x1b, y = A0y1b, type = "l", lwd = 3, col = "grey");

points(A1_r[1:100], A1_i[1:100],pch = 4, cex = 0.7);

text(x = -3.5, y = 2.25, labels = "b", cex = 2);
mtext(side = 1, "Real", outer = TRUE, line = 3, cex = 2);
mtext(side = 2, "Imaginary", outer = TRUE, line = 2.5, cex = 2);
```



To find out how frequently M was stable given that all $\gamma = 1$ versus $\gamma = \{1.95, 0.05\}$, the function below was created.

```
stab_bgamma <- function(S = 200, C = 0.05, Osd = 0.4, iters = 10000){
    ress     <- matrix(data = 0, nrow = iters, ncol = 2);
    A0_count <- 0;
```

```
    A1_count <- 0;
    while(iters > 0){
        A_dat  <- rnorm(n = S * S, mean = 0, sd = Osd);
        A_mat  <- matrix(data = A_dat, nrow = S);
        C_dat  <- rbinom(n = S * S, size = 1, prob = C);
        C_mat  <- matrix(data = C_dat, nrow = S, ncol = S);
        A_mat  <- A_mat * C_mat;
        gammas <- c(rep(1.95, S/2), rep(0.05, S/2))
        mu_gam <- mean(gammas);
        diag(A_mat) <- -1;
        A1     <- gammas * A_mat;
        A0     <- mu_gam * A_mat;
        A0_e   <- eigen(A0)$values;
        A0_r   <- Re(A0_e);
        A0_i   <- Im(A0_e);
        A1_e   <- eigen(A1)$values;
        A1_r   <- Re(A1_e);
        A1_i   <- Im(A1_e);
        if(max(A0_r) < 0){
            ress[iters, 1] <- 1;
            A0_count       <- A0_count + 1;
            }
        if(max(A1_r) < 0){
            ress[iters, 2] <- 1;
            A1_count       <- A1_count + 1;
        }
        print(c(iters, A0_count, A1_count));
        iters <- iters - 1;
    }
    return(ress);
}
```

The function above was run for `iters = 1000000`, and the resulting matrix `ress` was returned. Each row of `ress` represents a single M given $\gamma = 1$ (column 1) versus $\gamma = \{1.95, 0.05\}$ (column 2). Values of 0 indicate that M was found to be unstable (at least one real component of its eigenvalues greater than or equal to zero), whereas values of 1 indicate that M was found to be stable (all real components of eigenvalues are negative). The frequencies of stable M were 0 given $\gamma = 1$ and 0 given $\gamma = \{1.95, 0.05\}$, as reported in the main text and legend of Fig. 1.

Code and simulations underlying Fig. 2

Figure 2 of the main text shows how eigenvalue distributions in a system where $S = 1000$, $C = 1$, and $\sigma = 0.4$. Eigenvalues can be reproduced using the code below for when $\gamma = 1$ (panel a) and $\gamma \sim \mathcal{U}(0, 2)$ (panel b).

The code below reproduces the figure itself.

```
par(mfrow = c(1, 2), mar = c(0.5, 0.5, 0.5, 0.5), oma = c(5, 5, 0, 0));
plot(A0_r, A0_i, xlim = c(-16.5, 15.5), ylim = c(-16.5,15.5), pch = 4, cex = 0.7,
     xlab = "", ylab = "", cex.lab = 1.5, cex.axis = 1.5, asp = 1);
vl <- seq(from = 0, to = 2*pi, by = 0.001);
x0 <- sqrt(1000) * sd(A0vec) * cos(vl) + mean(diag(A0));
y0 <- sqrt(1000) * sd(A0vec) * sin(vl);
x1 <- sqrt(1000) * sd(A1vec) * cos(vl) + mean(diag(A1));
```

```
y1 <- sqrt(1000) * sd(A1vec) * sin(vl);
text(x = -15.5, y = 19, labels = "a", cex = 2);
points(x = x0, y = y0, type = "l", lwd = 3);
points(x = x1, y = y1, type = "l", col = "red", lwd = 3, lty = "dashed");
plot(A1_r, A1_i, xlim = c(-16.5, 15.5), ylim = c(-16.5,15.5), pch = 4, cex = 0.7,
     xlab = "", ylab = "", cex.lab = 1.5, cex.axis = 1.5, asp = 1, col = "red",
     yaxt = "n");
text(x = -15.5, y = 19, labels = "b", cex = 2);
points(x = x1, y = y1, type = "l", col = "red", lwd = 3)
points(x = x0, y = y0, type = "l", lwd = 3, lty = "dashed");
mtext(side = 1, "Real", outer = TRUE, line = 3, cex = 2);
mtext(side = 2, "Imaginary", outer = TRUE, line = 2.5, cex = 2);
```