Return Maps and Chaos in 2-Bus Shuttle Loop

Lukas WinklerPrins*

May 5 2014

Abstract

Using a nonlinear map model created by Dr Takashi Nagatani, we reevaluate headway of a two-bus shuttle system through a variety of graphical approaches. Specifically, we focus on characterizations of headway return map patterns for certain ranges of our speedup and loading parameters.

1 Introduction

This paper builds on the work of Dr Takashi Nagatani at Shizuoka University, specifically his 2006 paper "Chaos control and schedule of shuttle buses" [1]. The problem consists of two buses in a two-stop one-way bus loop (for instance, small shuttle loops). One stop accumulates passengers travelling to the other. Buses will speed up to make up for time spent loading and unloading passengers. Our goal is to analyze the headway, or time between buses, in the loop and observe where the headway becomes chaotic.

2 Model

2.1 Assumptions

There are a few key assumptions in our model:

1. Buses may freely pass each other.
2. Buses all share the same unadjusted speed.
3. Passengers each take a fixed time to board, and another fixed time to unload.
4. Passengers arrive at the boarding stop at a steady rate.

*lukas@brown.edu

5. Buses speed up on their route proportionally to the time taken at load and unload.
6. Buses have enough capacity to carry any number of passengers.

2.2 Nondimensionalization

I will briefly echo Nagatani's creation of a nondimensionalized map. We consider the number of passengers boarding bus i for trip m (in a total route/tour of M trips) $B_i(m)$. The parameter γ the time, per person, to board the bus, and η is the time it takes one person to exit the bus. Assuming the route is the same length, L in either direction, and $V_i(m)$ is the mean speed of bus i at trip m, then the time for a bus to complete one whole trip is $2L/V_i(m)$. The tour time, cumulative, is thus:

$$t_i(m+1) = t_i(m) + (\gamma + \eta)B_i(m) + \frac{2L}{V_i(m)} \tag{1}$$

We define μ as the rate at which passengers arrive at the boarding stop. So $B_i(m)$, the number of passengers boarding, is expressed by

$$B_i(m) = \mu(t_i(m) - t_{i'}(m')) \tag{2}$$

where bus i' on trip m' is the bus that arrived just before bus i. So that $B_i(m)$ is the number of passengers who have arrived since the bus just ahead left. In our two-bus system, this might be bus 1 or bus 2; it depends on the exact dynamics. This facet of our model complicates what would otherwise be a simple two-dimensional map system.

Buses speed up proportionally to the delay at the stops, by a speed up factor s_i. Thus

$$V_i(m) = V_0 + s(i(\gamma + \eta)B_i(m) \tag{3}$$

with V_0 as the mean bus speed for both buses. Through combining equations 1-3, we find:

$$t_i(m+1) = t_i(m) + \mu(\gamma+\eta)(t_i(m) - t_{i'}(m')) + \frac{2L}{V_0 + s_i\mu(\gamma+\eta)(t_i(m) - t_{i'}(m'))} \tag{4}$$

We define our nondimensional time, loading parameter, and speedup parameters as $T_i(m) = t_i(m)V_0/2L$, $\Gamma = \mu(\gamma + \eta)$, and $S_i = s_i\mu(\gamma + \eta)2L/V_0^2$. Thus our dimensionless nonlinear map is:

$$T_i(m+1) = T_i(m) + \Gamma(T_i(m) - T_{i'}(m')) + \frac{1}{1 + S_i(T_i(m) - T_{i'}(m'))} \tag{5}$$

This applies for both buses in our 2-bus system, $i = 1, 2$. In the original paper, Nagatani does not define "headway," but we choose to define it as:

$$H_i(m) = |T_i(m) - T_j(m)| \tag{6}$$

which reflects a definition used by Nagatani in a similar paper from 2002 [2]. This quantity can diverge quickly if one bus "laps" another frequently.

2.3 Analysis

The analysis Nagatani applies in a paper with a simpler map [2] cannot apply here due to the incorporation of bus speed up and slow down in our model, thus a formula for a fixed headway cannot be found precisely.

We can generally anticipate higher S1, S2 values as keeping diverging headway at bay, until Γ becomes too large.

With bus headway as our chief concern, we anticipate a few cases. When S1 and S2 are the same or similar, the buses may fall into sync with $H(m)$ and $\Delta T_i(m)$ constant.

$H(m)$ might also be periodic, with the headway shrinking and growing. In a 2-period scenario of this, bus 1 might arrive at the origin with bus 2 following at some small $H(m)$. If we consider this as being similar to the trip $m-1$, then there will be a build-up of people at the stop and bus 1 will speed up quite a bit. Due to the small headway, bus 2 will not speed up much in its loop m and the headway will grow to $H(m+1) > H(m)$. Now, relatively, due to bus 1's speed-up, not much time has passed since the last bus arrival ($T_{i'}(m') = T_2(m)$ in this case) at the origin, so bus 1 will slow down for trip $m+1$ for lack of passengers. Bus 2, on the other hand, will be lagging behind bus 1 such that it will speed up when it picks up the build-up of passengers for trip $m+1$, with $T_{i'}(m') = T_1(m+1)$. And thus $H(m+2) < H(m+1)$. For the right windows of Γ, S1, and S2, $H(m+2) = H(m)$. One can generalize this thought process to understand n-period orbits.

For other nearby parameter ranges, the headway values might become periodic, pseudo- or loosely periodic, increase towards infinity, or bounce around an interval of possible values. We seek to explore this in our return maps.

When S1 and S2 are not close to each other, we anticipate more buses passing of one another and divergence of the headway. Nagatani corroborates with this in both the 2006 paper [1], and two slightly different models from 2002 papers [2] [3].

3 Computation

We explore the bus behavior with time through computer simulation in MATLAB. We seek the long-term behavior and found through trials that transient

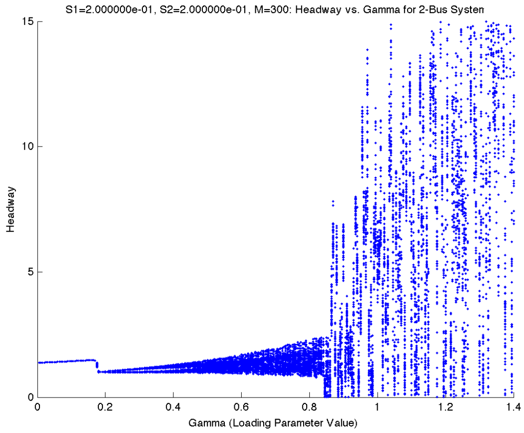

Figure 1: Test simulation with last 100 points displayed, M=300

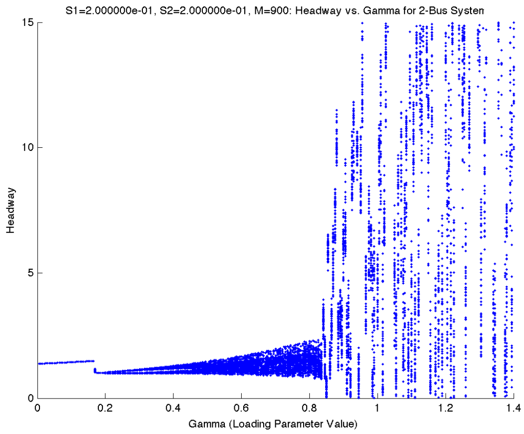

Figure 2: Test simulation with last 100 points displayed, M=900

headway values due to initial conditions vanish around $m = 200$. Thus, to plot, we run $M = 300$ loops for each bus and then plot only the last 100 leger times for each bus.

The marginal differences in Figures 1, 2 indicates a run of $M = 300$ is sufficient to avoid transient bus motion as per initial conditions. The program is run using 4 code samples (included at end) and takes a few minutes to run in its current state for $M = 300$. The code executes as such:

1. Values of S1, S2, and M are set by hand.
2. A loop begins going through values of Γ from 0 to 1.4 (in our code, in increments of 0.005). As per Nagatani's paper, $\Gamma > 2$ clearly diverges. For all of my tests, divergence came even earlier.
3. For each value of gamma, a leger of bus times is produced through our

map. The leger contains the bus arrival times in order, with corresponding bus number and loop number at each time.

4. The leger is clipped to the last 100 trips and parsed between bus 1 and bus 2 for calculation of headway.

5. The results are plotted in the desired format with some additional data formatting for each gamma value.

"Bus_Scheduling.m" is the only function that must be run through the MATLAB interface (or command line). It calls "buses_function.m" to populate the leger, which uses "TIncrement.m" to do the map jumps from m to $m+1$ and "make_io_matrix.m" for some matrix formatting. See the code for further details.

4 Results

S values beyond 1 are relatively predictable: the buses catch up quickly enough to keep a constant headway up until Γ becomes very large. Thus we focus our discussion on $0 \leq S \leq 1$. For all values tested, bus headway became chaotic as $\Gamma > 1.4$, approximately, whereas for Nagatani put a cut-off at $\Gamma > 2$.

4.1 Headway Orbits

We plot a variety of Headway vs. Gamma plots (orbit diagrams) for different combinations of S1, S2 values in Figure 3.

Given our symmetric definition of headway, that maps are similar for interchanged values of S1 and S2 is expected. Note that the maps in the upper right, lower left corners (plots C, D, H, I, M, and N) have much larger scales for headway than the other 10 plots. This is to be expected as well: when one bus has high speedup values compared to the other, the headway increases rapidly.

Most diagrams move between n-periodic orbits, coverings of a range of headway values, and chaotic motion (and sometimes back and forth between these types). In the diagrams, these look like 1D curves, vertical bands, and dispersed points, respectively.

In Figure 4, until $\Gamma \approx 0.23$ there is one stable headway value ≈ 1. Past $\Gamma = 0.23$, the headway drops and splits into two, viewable in Figure 5. Somewhere between $\Gamma = 0.5$ and $\Gamma = 0.6$ the two branches join and we have the "banding" behavior, where $H(m)$ stays within a range. With $\Gamma = 0.4$, we plot $H(m)$ vs m in Figure 6 to see an "echo" nature of the bus moving in the two bands: the buses keep a similar headway for a few steps, then jump to the upper band with a higher headway, and then back down to repeat this process. Given the repeating nature in Figure 6 this might be a high-level periodic orbit. With two "bands" in the orbit diagram, it is a pseudo-2-period orbit. We differentiate "banding" from chaos by its limited range of $H(m)$ values and predictability—later in this paper,

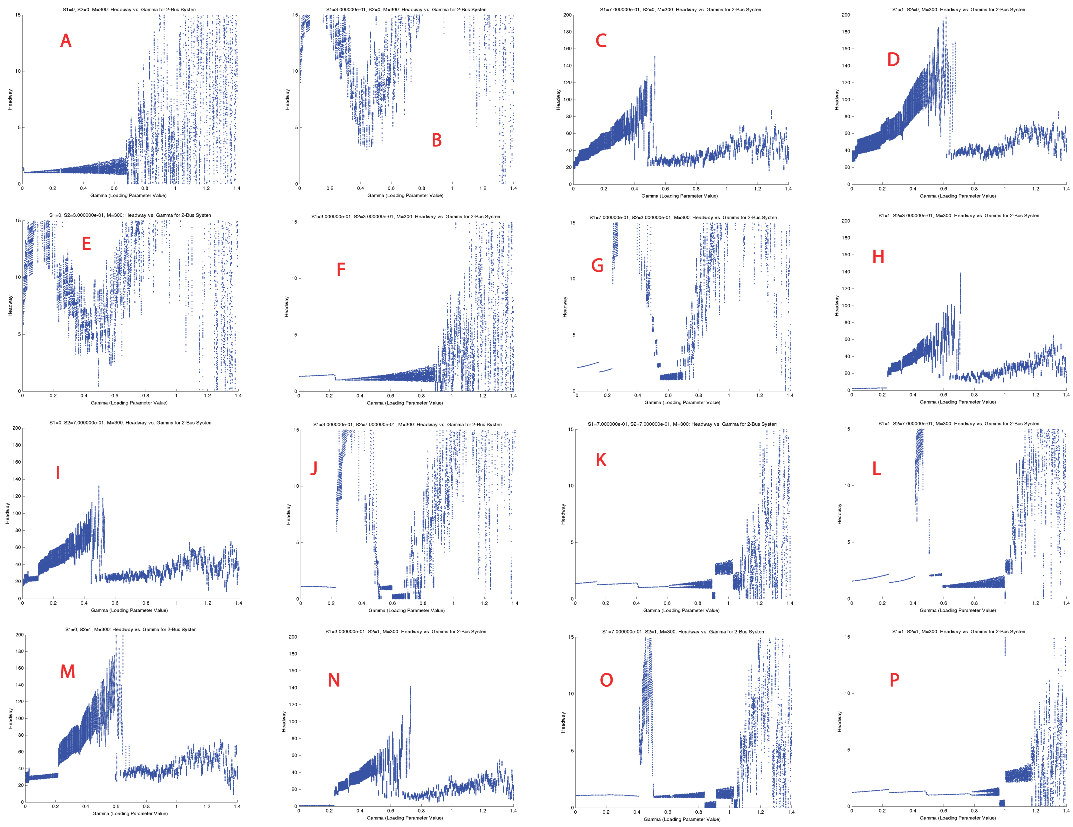

Figure 3: Grid of headway orbit diagrams, $H(m)$ vs Γ. From left to right, S1 values are 0, 0.3, 0.7, 1. The same top to bottom for S2 values. All for M=300.

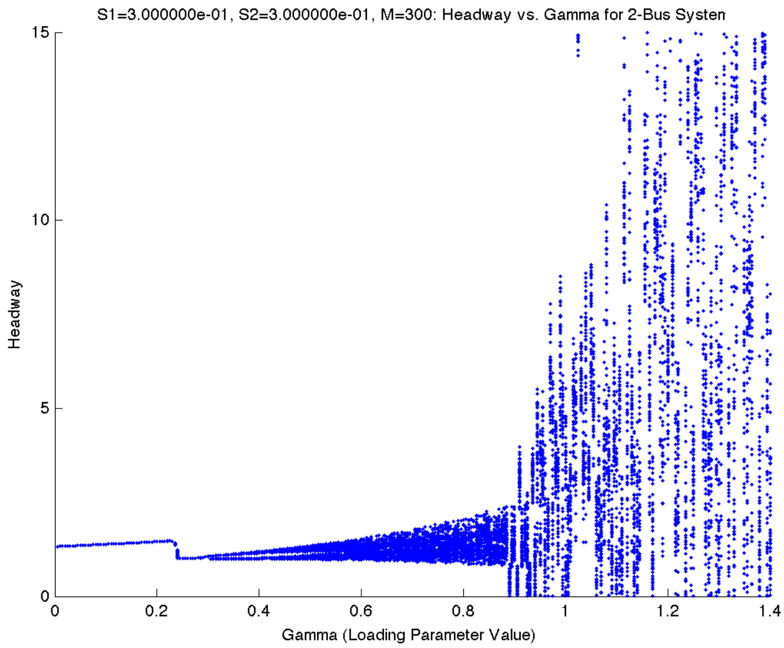

Figure 4: S1=S2=0.3 for M=300 Orbit Diagram.

in our return map analysis, clarity between $H(m)$ values that periodically or linearly change are easy to see compared to chaotic variation.

Beginning with $\Gamma \approx 0.885$ chaotic behavior begins—$H(m)$ can reach 0 due to large speed-up behavior for the buses. Figure 7 makes it clear that there is an erratic but directed motion in the increasing $H(m)$ vs. m. For other $\Gamma > 0.885$ values, the trends in the plots is unpredictably positive or negative, with the range span growing more or less with Γ. Another is provided for comparison in figure 8.

Figures 7 and 8 help affirm that in our chaotic regions of Γ the headway still varies *relatively* smoothly, though unpredictably as m increases; our return plots yield insights.

We regularly see jumps in the orbit diagram at $\Gamma \approx 0.23$. The bottom right four plots in Figure 3 also indicate values of $\Gamma \approx 0.41$ and $\Gamma \approx 0.51$ are recurrent bifurcation locations, as visible in plots G, J, K, L, O, and P. The nature of the bus headway might change, e.g. from steady to dramatic banding at $\Gamma \approx 0.41$ in plot O. Or, there might be a small shift, e.g. at $\Gamma \approx 0.51$ in plot P, from one stable periodic $H(m)$ to another. By plotting more parameter values, e.g. $S = S1 = S2 = 0.5$ in Figure 9, we see that these values of Γ are likely not unique: increasing S moves the bifurcations towards higher Γ values (in addition to introducing some new behavior). The onset of chaos is also "pushed back" as S increases, visible in the diagonal plots in Fig. 3 (A, F, K, P) and Fig. 9. This effect is consistent even when $S > 1$.

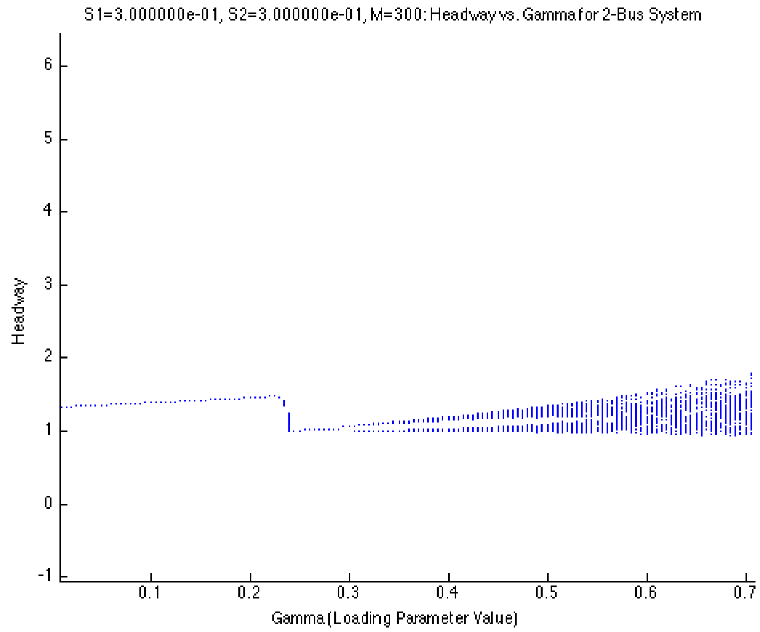

Figure 5: S1=S2=0.3 for M=300 Orbit Diagram, enhanced near $\Gamma = 0.23$.

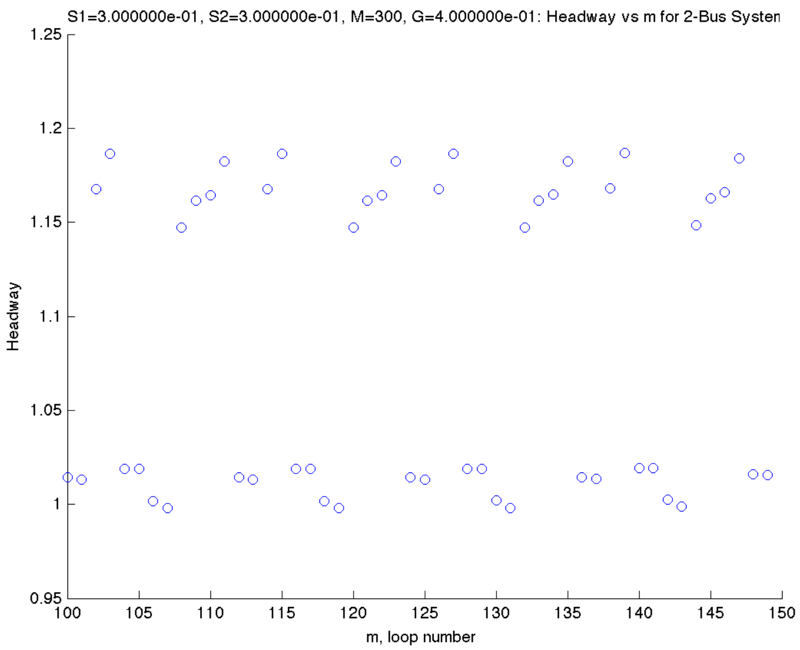

Figure 6: S1=S2=0.3 for M=300 $\Gamma = 0.4$ Headway vs. loop m.

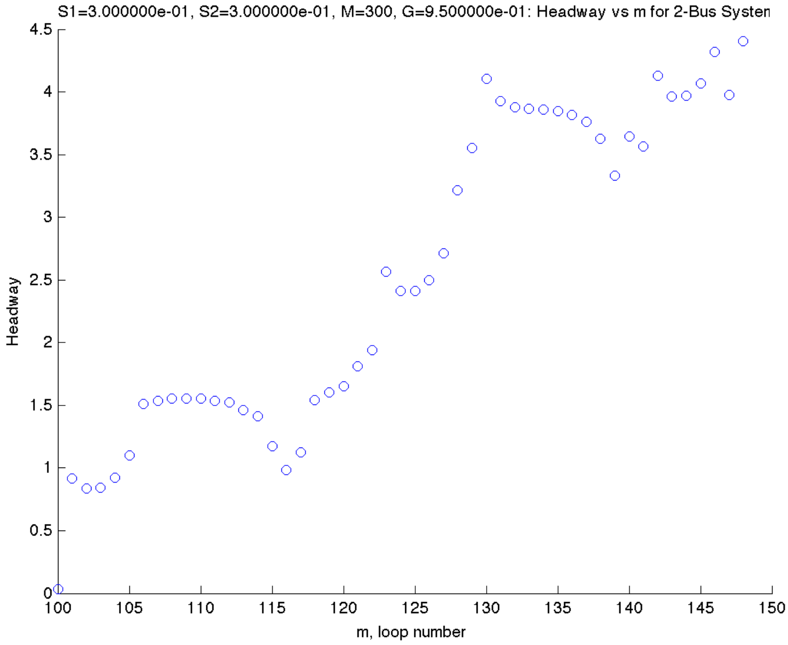

Figure 7: S1=S2=0.3 for M=300 Γ=0.95 Headway vs. loop m.

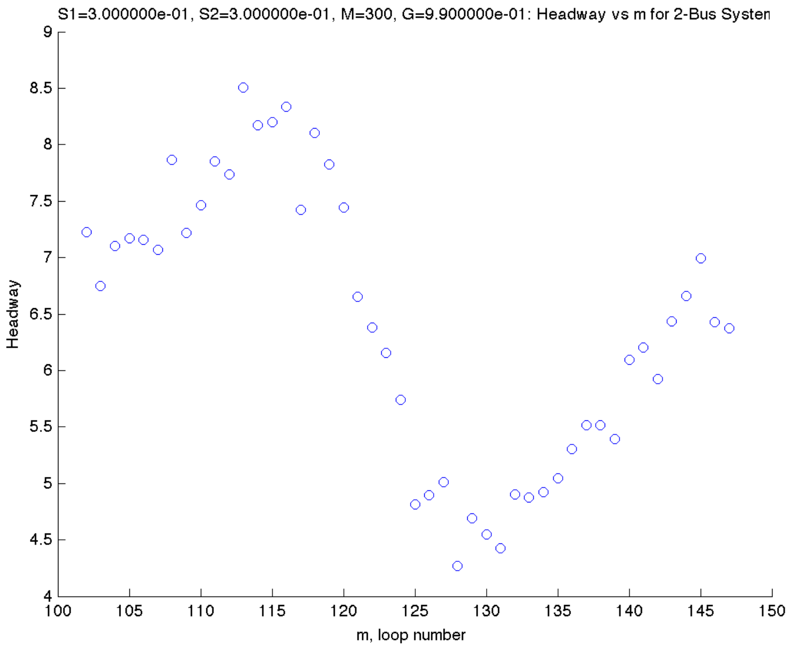

Figure 8: S1=S2=0.3 for M=300 Γ=0.99 Headway vs loop m.

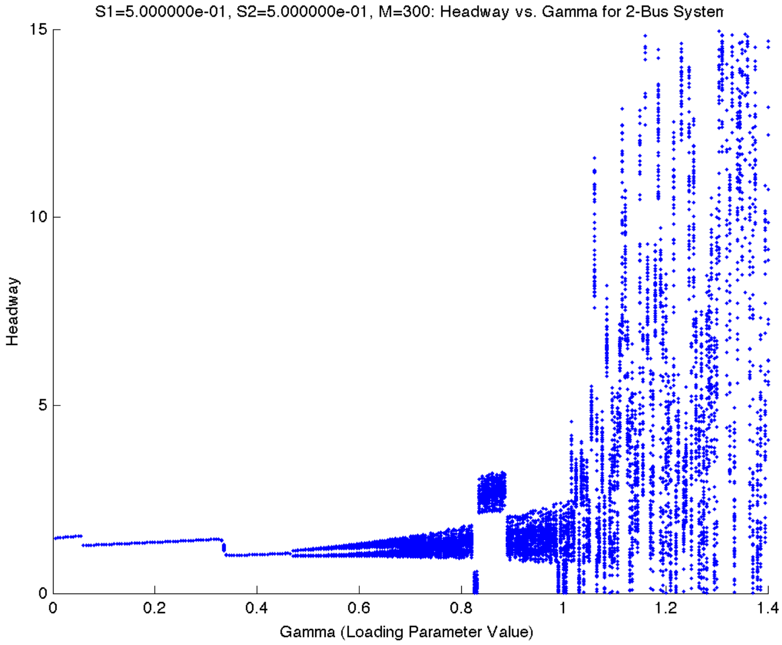

Figure 9: S1=S2=0.5 for M=300 Orbit Diagram.

4.2 Headway Return Map Types

Some of our most interesting results come from studying the "return maps" of Headway values, i.e. plots of $H(m+1)$ vs $H(m)$. This gives us a more intuitive understanding of the bus behavior for a specific loading parameter or Γ value. For a variety of parameter values, there emerge about 6 varieties, though these may be intertwined.

The simplest is the "point". This occurs at low Γ values, where $H(m)$ is fixed. The point falls on the line $H(m+1) = H(m)$. No plot is provided as it is an uninteresting case in the steady-state.

A very common case is the "rough line," seen in Figure 10. We know the direction to follow from Figure 7—$H(m)$ steadily works its way up, though it has "ghost" fixed points near the line $H(m+1) = H(m)$. These correspond to the sequences of very similar headway values in the intervals $105 < m < 115$ and $130 < m < 140$, approximately.

Another case I call the "creep"—seen in Figure 11. In this map, the headway "creeps" along the line $H(m+1) = H(m)$, always close to being a fixed point but steadily growing. As with all of these cases we have not found exactly what parameter ranges they occur in, but creep's connection to our next case might give some insight.

When S1 and S2 are very different, and for a low Γ, there is a "step" motion in the headway return plot, as in Figure 12. Once again, our value of $H(m)$

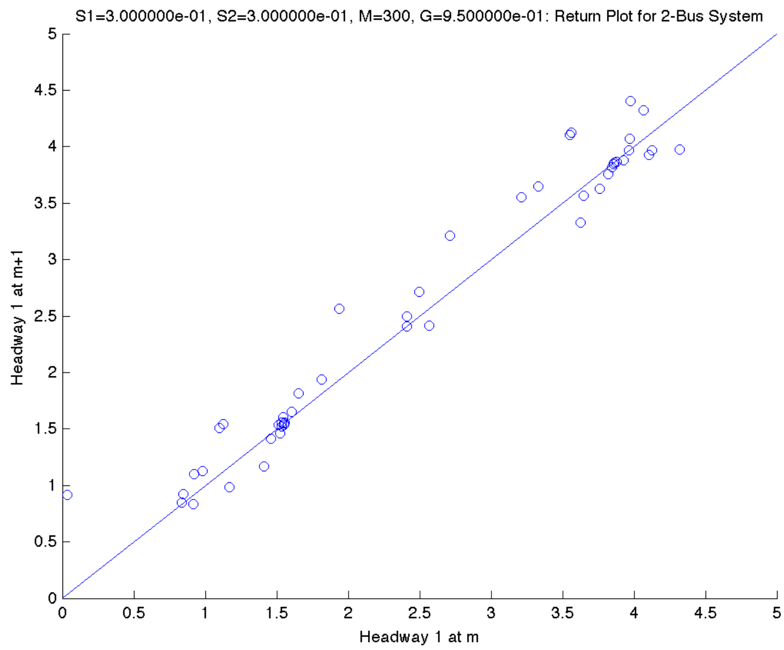

Figure 10: "Rough LIne". S1=S2=0.3 for M=300 and Γ=0.95 Return Map.

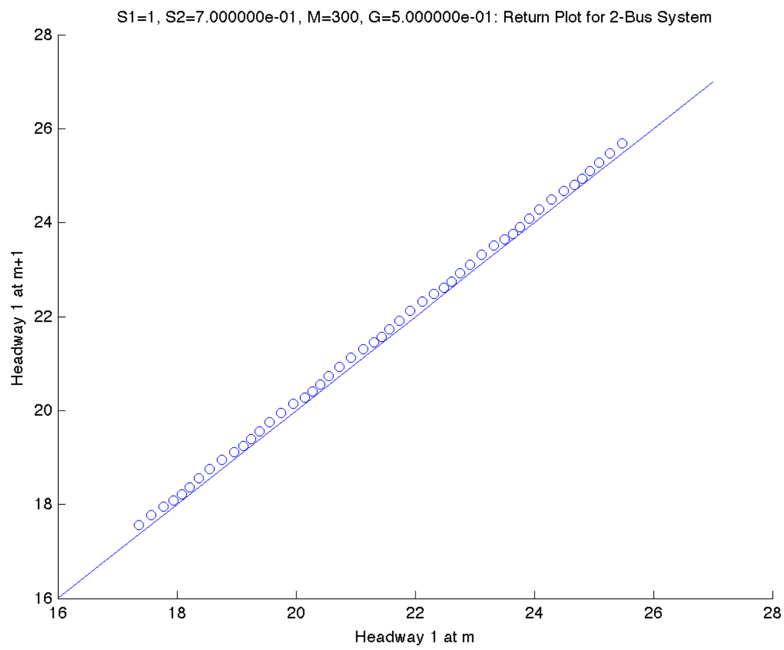

Figure 11: "Creep". S1=1, S2=0.7 for M=300 and Γ=0.5 Return Map.

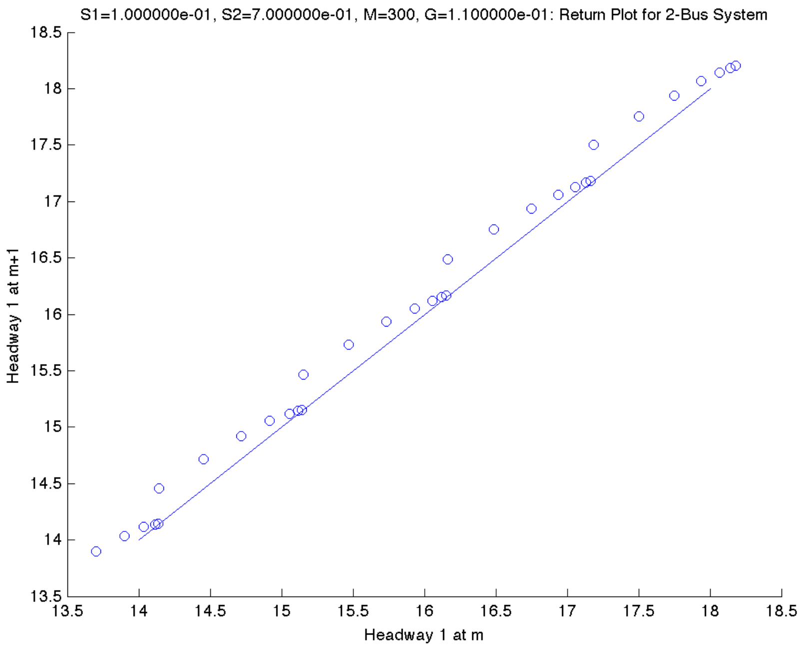

Figure 12: "Steps." S1=0.1 S2=0.7 for M=300 and Γ=0.11 Return Map.

increases with time, but now in a non-steady but predictable way. We conjecture that the ghost fixed points occur when the fast-moving bus happens to follow the slow-moving bus very closely, so that it does not speed up almost at all while the slow-moving bus attempts to close the gap in interval time. But by the time it gets around, the fast-moving bus has taken off again. This pattern repeats when the fast-moving bus laps the slow-moving bus. The pattern in Figure 12 is reflected over the $H(m+1) = H(m)$ line if the S values are reversed. This effect happens for S1=0.1, S2=0.7 in a range of $0.095 < \Gamma < 0.195$. As Γ increases, the effect flattens and we approach the "creep" case. This evolution will be returned to in the next subsection.

One of the more curious cases, the "V", occurs in similarly narrow ranges of parameters. This corresponds to the "banding" behavior discussed earlier—where $H(m)$ stays within a range for all m. The return plot is Figure 13. Near where it crosses the line $H(m+1) = H(m)$ there is close to a 2-period loop. Figure 14 shows that $H(m) < 0.6$ during the full bus tour, and it returns frequently near ≈ 0.21, only to then jump around an interval about $H(m) = 0.21$.

The final case we consider distinct we call the "catamaran" or "comets." In it, there are three or four streaks of points, as visible in Figure 15. It reflects the "echo" effect we saw in Figure 6, where the "pontoons" (to drag out the catamaran reference) on either side of the $H(m+1) = H(m)$ line send orbits back and forth between each other. It is common when S and Γ balance out—i.e. when S and Γ are either both large or both small. In many instances

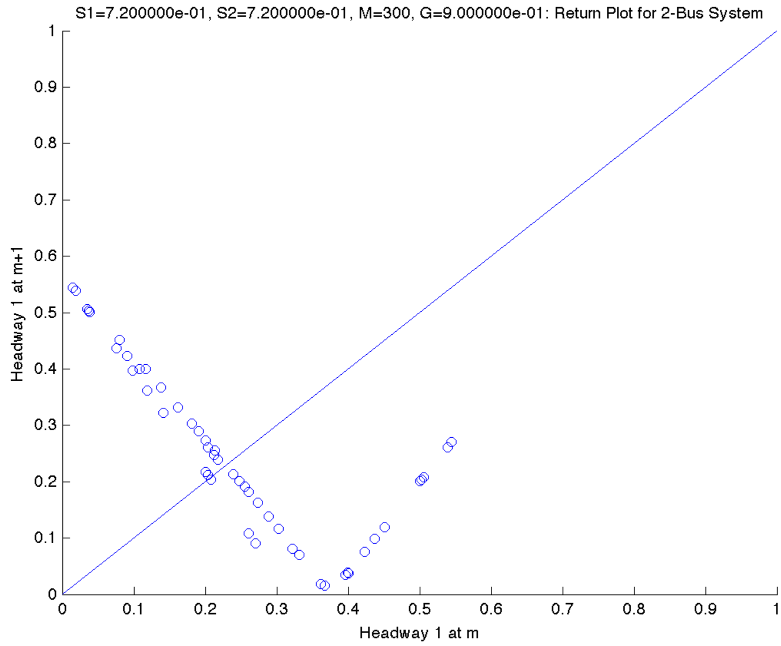

Figure 13: "V". S1=S2=0.72 for M=300 and Γ=0.9 Return Map.

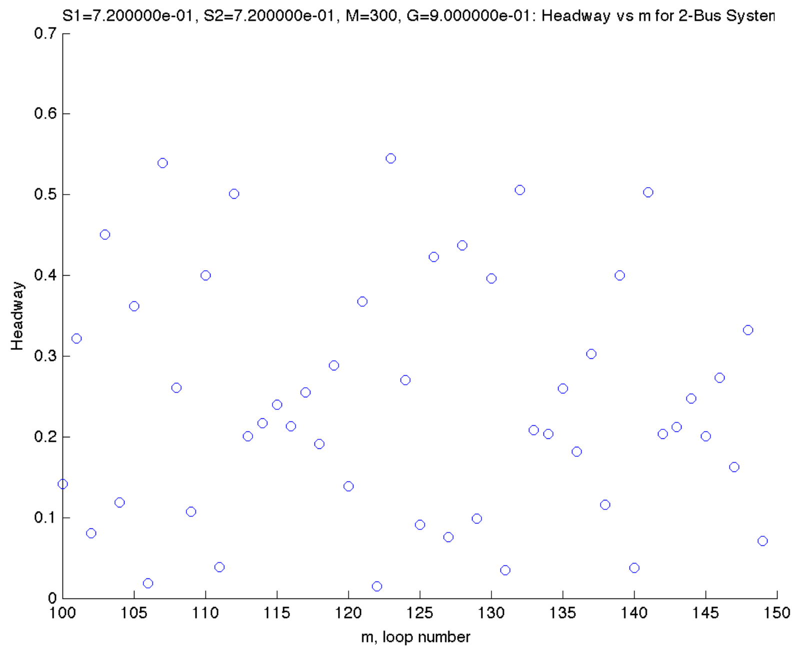

Figure 14: S1=S2=0.72 for M=300 Γ=0.9 Headway vs. loop m. Headway of the "V"-type return plot with increasing loop number m.

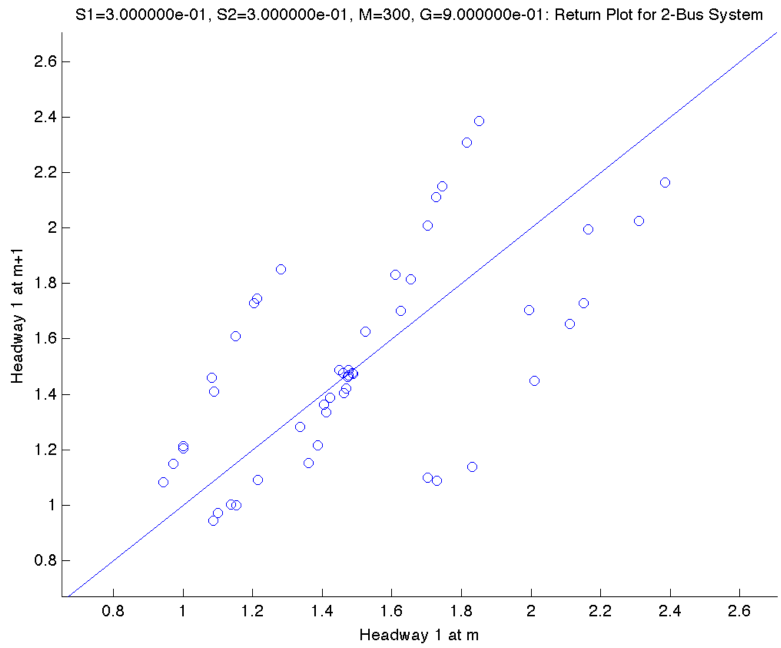

Figure 15: "Catamaran". S1=S2=0.3 for M=300 and Γ=0.9 Return Map.

"catamaran" blends into looking like a "rough line."

4.3 Return Map Evolution

This last point on catamarans and the steps turning into creeps led me to investigate how the return plots change with Γ. This links them to the headway plots in our orbit diagrams. As the S1=S2 cases evolve in a more regulated way, I investigated two alternative cases: S1=0.1 and S2=0.7; and S1=1.0 and S2=0.7. In the first, as Γ is increased from $0 < \Gamma < 1.4$, we see four cases emerge:

1. "Point" in $0 < \Gamma < 0.09$.
2. "Steps" in $0.09 < \Gamma < 0.2$
3. "Creep" in $0.2 < \Gamma < 0.55$
4. "Rough Line" for $\Gamma > 0.55$

Where all values of Γ are approximate. In the second case (where S1=1.0 instead of S1=0.1):

1. "Point" in $0 < \Gamma < 0.42$.

2. "Steps" in $0.42 < \Gamma < 0.48$
3. "Creep" in $0.48 < \Gamma < 0.51$
4. "Catamaran" in $0.51 < \Gamma < 1$.
5. "V" in $1 < \Gamma < 1.01$
6. A very scattered "catamaran" for $\Gamma < 1.01$
7. And as Γ gets larger, the "catamaran" becomes a "rough line".

Given the tiny window "V" behavior occurs in (in the second case), and the blending of "catamaran" and "rough line", it is possible that the first case does not detect this type of behavior and does not differentiate between "catamaran" and "rough line" at our level of fidelity.

With this evolution, we see how a stable $H(m)$ value in a point can suddenly break into a headway that is consistently getting close to stable, but thrown off due to a subtly larger loading parameter. This ability to get very close to stable degernerates as Γ increases further, and the headway evolves into a creep type. Within some windows, we see the "echo" periodic effect of catamarans and V-type headway diagrams. As we approach a diverging $H(m)$, the catamaran pontoons merge and a chaotic but positively sloped rough line emerges, indicating chaotic diverging headway. What is interesting to see is the creep behavior, of a diverging $H(m)$ value, settles back to a limited range as a catamaran in the second case, before diverging as a rough line.

5 Further Work

There is a lot of work that could still be done here. Some further areas of investigation include:

1. Can we take cross-sections of our chaotic regions and look for patterns such as the Cantor Set?
2. Can we more precisely define where chaos begins through Liapunov exponent analysis?
3. A more wholistic view of where the different return map types occur—i.e. testing of all parameter combinations will add to completeness of the theory.
4. In Figure 3, in the plots C, D, H, I, M, N (where S1 and S2 are most different), why do the headway values begin to look like chaotic divergence, but then come back down near $\Gamma = 0.6$? I suspect this has something to do with "jamming" in certain speedup and loading parameter ranges: once Γ increases enough the stream of people at the stop so that both buses have steadier motion, as opposed to bus arrival heavily impacting tour times.
5. Analysis of individual bus tour time is entirely absent from this paper and might yield information about bus behavior.

6 Conclusion

We affirm conclusions in Nagatani's 2006 paper [1] and another student-made follow up at the University of Arizona [4] that the speedup and loading parameters together regulate the onset of chaos. We also conclude that headway goes through a diverse array of long-term behaviors as Γ changes, which connect our return map analysis to our headway orbit diagrams. Headway may begin to diverge for some ranges of Γ but then return to stable values due to an as-of-yet unexplained effect, but predicted by our return map evolution.

References

[1] T. Nagatani, *Chaos control and schedule of shuttle buses*, Physica A, 371 (2006), pp. 683–691.

[2] T. Nagatani, *Chaos and headway distribution of shuttle buses that pass each other freely*, Physica A, 323 (2003), pp. 686–694.

[3] T. Nagatani, *Dynamical transitions to chaotic and periodic motions of two shuttle buses*, Physical A, 319 (2003), pp. 568-578.

[4] M. Avetian, L. Guzman, R. Kumada, R. Soimaru, S. Zerkoune, *Chaos Control in Shuttle Bus Schedules*, 2010. Web.

7 Code

7.1 Bus_Scheduling.m

```
% Lukas WinklerPrins
% APMA 1360 Gemmer Spring 2014
% Research Project on Bus Scheduling & Chaos

gam_incr = 0.005;
gam_end = 1.4; % beyond this we see only chaos in aperiodicity.
gam = gam_incr:gam_incr:gam_end;

% Set these values by hand:
S1 = 0.3;
S2 = 0.3;
M = 300;
% 300 seems to be enough to escape transients from ICs

figure
hold on
cutoff = '';

%for ig = 1:length(gam)
for ig = 180 % used when evaluating a specific gamma value.

    leger = buses_function(gam(ig),S1,S2,M);

    leger = leger((M-100):M,:); % just the last 100 rows of leger necessary.
    cutoff = 'cutoff';

% SEPARATE BUS 1 AND BUS 2 ///////////////////////////////////

% leger is three columns: TIME, BUS, and that bus' TRIP NUMBER
    bus_one_stops = leger(:,2) == 1;
    bus_two_stops = leger(:,2) == 2;
    bus_one = [];
    bus_two = [];

    for ii = 1:length(leger)
        if(bus_one_stops(ii))
            bus_one = [bus_one; leger(ii,:)];
        end
        if(bus_two_stops(ii))
            bus_two = [bus_two; leger(ii,:)];
        end
    end

% FINDING HEADWAY ////////////////////////////////////////////

    headway_start = max(min(bus_one(:,3)),min(bus_two(:,3)));
    headway_end = min(max(bus_one(:,3)),max(bus_two(:,3)));
    diff = headway_end-headway_start;

    b1s = find(bus_one(:,3)==headway_start);
    b2s = find(bus_two(:,3)==headway_start);
    % make sure buses legers line up correctly.
    headway = abs(bus_one(b1s:b1s+diff,1) - bus_two(b2s:b2s+diff,1));

% PLOTTING //////////////////////////////////////////////////

    % PLOTTING HEADWAY
%     gamma_vec = gam(ig)*ones(size(headway));
%     scatter(gamma_vec,headway,'b.');
%     y_string = sprintf('Headway');

    % PLOTTING RETURN MAPS

    h = length(headway);
    scatter(headway(1:(h-1),1),headway(2:h,1));
    lilgam = gam(ig);

    % PLOTTING VS M
```

```
    % headway vs m
%      scatter(headway_start:headway_end,headway);
%      lilgam = gam(ig);
%      title_string = sprintf('S1=%d, S2=%d, M=%d, G=%d: Headway vs m for 2-Bus System',S1,S2,M,lilgam);
%      ylabel('Headway');

      % DeltaT vs m
%      l = length(bus_one);
%      scatter(1:(l-1),bus_one(2:l,1)-bus_one(1:(l-1),1));
%      lilgam = gam(ig);
%      title_string= sprintf('S1=%d, S2=%d, M=%d, G=%d: Delta T vs m for 2-Bus System',S1,S2,M,lilgam);
%      ylabel('Delta T1');

end

% For plotting headway vs gamma:

% axis([0, gam_end,0,15]);
% %axis([0 1.4 0 200]);
% xlabel('Gamma (Loading Parameter Value)');
% ylabel(y_string);
% title_string = sprintf('S1=%d, S2=%d, M=%d: %s vs. Gamma for 2-Bus System',S1,S2,M,y_string);
% title(title_string);
%
% print('-dpng',sprintf('%d_%d_M%d_%s.png',S1,S2,M,cutoff));

% And for plotting return maps:

xlabel('Headway 1 at m');
ylabel('Headway 1 at m+1');
title_string = sprintf('S1=%d, S2=%d, M=%d, G=%d: Return Plot for 2-Bus System',S1,S2,M,lilgam);
title(title_string);
print('-dpng',sprintf('%d_%d_M%d_%s_gam%d_returns.png',S1,S2,M,cutoff,lilgam));

% And for plotting vs m:

% xlabel('m, loop number');
% title(title_string);
% print('-dpng',sprintf('%d_%d_M%d_%s_gam%d_headway.png',S1,S2,M,cutoff,lilgam));
%
% print('-dpng',sprintf('%d_%d_M%d_%s_gam%d_tourtime.png',S1,S2,M,cutoff,lilgam));
```

7.2 buses_function.m

```
function [leger,temp_recent_bus] = buses_function(gamma,S1,S2,M)

% Lukas WinklerPrins
% APMA 1360 Gemmer Spring 2014
% Research Project on Bus Scheduling & Chaos

start1 = 1;
start2 = 2.5;
% These values are fairly arbitrary... given we only plot long-term.
% I tried a variety of values < 5 and no difference in the long-term.

leger = [0 1 0; 0 2 0; start1 1 1; start2 2 1];
% time, bus, trip #.
% We initialize with two "trips" starting from zero.

m1 = 2;
m2 = 2;

while(m1<M || m2<M)

    while(1) %%%%%%%%%%%%%%%%%%%%%% BUS ONE
        temp_recent_bus = make_io_matrix(find(leger(:,2)==1),length(leger))*leger(:,1);
        % See where bus 1 is in the leger. Turn this 1/0 vector into a
        % matrix with the 1/0 along the diagonal, multiply it by leger
        % to get a vector with 0 for bus 2 times, and the correct times for
        % bus 1. Now we can max this to find the location of the highest
        % bus 1 time...

        % Remember that find gives INDEX, not VALUE
        % We also only need the index from max, not the value.
        [~,recent_bus] = max(temp_recent_bus);
        T1 = leger(recent_bus,1);

        %[~,recent_bus] = max(leger(:,1).*(leger(:,2)==1);
        %T1 = leger(recent_bus,1);

        temp_recent_time = make_io_matrix(find(leger(:,1) < T1),length(leger))*leger(:,1);
        % now we are just looking for the time, from either bus, closest
        % to but less than T1
        % maybe should be <= ?
        [~,recent_time] = max(temp_recent_time);
        Tj = leger(recent_time,1);

        if(leger(recent_time,2) == 1)
            % Recent time is also bus 1

            T1p = TIncrement(T1,gamma,Tj,S1);
            leger = [leger; T1p 1 m1];
            m1 = m1 + 1;
            break;
            % if it needs another self-jump, bus two will send it back in the big while() loop

        else % jumping from bus 2's position

            % We have to make sure bus 2 cannot self-jump again before bus
            % 1 jumps from its recent time Tj.

            temp = make_io_matrix(find(leger(:,2)==2),length(leger))*leger(:,1);
            [~,recent_bus] = max(temp);
            temp_T = leger(recent_bus,1);

            temp_Tp = TIncrement(temp_T,gamma,Tj,S2);
            if(temp_Tp < T1)
                break;
                % Allow bus 2 to increment from itself before we do bus 1.
            end

```

```
            % Otherwise, increment from bus 2 as usual
            T1p = TIncrement(T1,gamma,Tj,S1);
            leger = [leger; T1p 1 m1];
            m1 = m1 + 1;
            break;

        end % if
    end % bus 1 while

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % All the above code is echoed for bus two.

    while(1) %%%%%%%%%%%%%%%%%%%%%% BUS TWO
        temp_recent_bus = make_io_matrix(find(leger(:,2)==2),length(leger))*leger(:,1);
        [~,recent_bus] = max(temp_recent_bus);
        T1 = leger(recent_bus,1);
        temp_recent_time = make_io_matrix(find(leger(:,1) < T1),length(leger))*leger(:,1);
        [~,recent_time] = max(temp_recent_time);
        Tj = leger(recent_time,1);

        if(leger(recent_time,2) == 2) % self-jumping.
            T1p = TIncrement(T1,gamma,Tj,S2);
            leger = [leger; T1p 2 m2];
            m2 = m2 + 1;
            break;
        else % jumping from bus 1's position
            temp = make_io_matrix(find(leger(:,2)==1),length(leger))*leger(:,1);
            [~,recent_bus] = max(temp);
            temp_T = leger(recent_bus,1);
            temp_Tp = TIncrement(temp_T,gamma,Tj,S1);
            if(temp_Tp < T1)
                break;
            end
            T1p = TIncrement(T1,gamma,Tj,S2);
            leger = [leger; T1p 2 m2];
            m2 = m2 + 1;
            break;
        end % if
    end % bus 2 while

end % big ol m while

end
```

7.3 make_io_matrix.m

```
function matrix = make_io_matrix(vector,size)
    % find gives INDEX... vector is of OK indexes in a vector.

    matrix = zeros(size);

    for ii = 1:length(vector)
        matrix(vector(ii),vector(ii)) = 1;
    end

end
```

7.4 TIncrement.m

```
function Tnplus1 = TIncrement(T1,gam,Tj,S)

Tnplus1 = T1 + gam*(T1-Tj)+1/(1+S*(T1-Tj));

end
```