Differential Equations Governing Traffic Patterns

Calix Barrus, Laren Edwards, Riley Gabrielson, Dallan Olsen, Elijah Cox

December 7, 2021

December 7, 2021

Abstract

Model the flow of traffic at a stoplight.

1 Governing Equations

We will use the equations

$$u_t + L(x,t)(1 - \frac{2n}{u_\infty})u_x = 0$$

and analyze this equation considering several cases and initial conditions. We have gotten this equation from the discussion on traffic in 8.1 of Volume 4, but have replaced the term V_∞ with $L(x,t)$ where $L(x,t)$ will be defined as to simulate different stop light patterns. Other variables in this equation include u_∞ which refers to the maximum density of cars at a given position.

We expect to define the light equation L as $L : \mathcal{R} \times \mathcal{R} \to [0, V_\infty]$. In situations where we are modeling cars getting going from stop as the light turn green, $L(x,t) = V_\infty$ and will be constant. In the situation where we model a red light turning on to stop oncoming traffic, we will examine different functions for $L(x,t)$ to see what kind of definition produces a realistic simulation of traffic flow.

2 Solutions

We will attempt to solve these equations analytically, but will likely end up using numerical methods, especially as we get into weirder initial conditions (such as cars stopping at a red light).

If we are successful at modeling cars starting and stopping at a light, we will attempt to model an intersection with cars coming and leaving in two perpendicular directions.

3 SIR Details

To better approximate a more complicated system with multiple interacting streams, we employed an SIR styled model.

Structure While there are many different traffic light intersections and variants, this model assumes a standard 4 way intersection with traffic coming from each of the cardinal directions

TrafficDiagrampng.png

(north, east, south, and west). Each direction has 3 streams of traffic, a turning left stream, a straight stream, and a right stream.

The change in the straight stream is only dependent on whether the light is red or green. The left stream is "unprotected", meaning that the change in the left stream depends on whether the light is green and how much traffic is coming from the opposite straight stream. The right stream is free to turn at any time, and is only dependent on how much traffic is coming from the perpendicular straight stream.

3.1   Implementation

First, we define a light function. We want this function to be periodic and to move between 0 (red) and 1 (green). Rather than having one function that returns red for one direction and green for the other direction, we created two correlated functions that define the state of the light in the north-south direction and the east-west direction:

$$\text{lightNS(t)} = \begin{cases} \sin(3t) & t \ \% \ 2\pi < \frac{\pi}{6}) \\ 1 & t \ \% \ 2\pi \in [\frac{\pi}{6}, \frac{5\pi}{6}) \\ \sin(3t) & t \ \% \ 2\pi \in [\frac{5\pi}{6}, \frac{\pi}{)} \\ 0 \text{ otherwise} \end{cases}$$

$$
\text{lightEW(t)} = \begin{cases}
\text{-sin(3t)} & t \ \% \ 2\pi > \frac{11\pi}{6}) \\
1 & t \ \% \ 2\pi \in (7\frac{\pi}{6}, 11\frac{5\pi}{6}] \\
\text{-sin(3t)} & t \ \% \ 2\pi \in (\pi, 7\frac{\pi}{)} \\
0 \ \text{otherwise}
\end{cases}
$$

Next, we define 12 different "bins", one for each stream of our model (eg. North Forward), and then define the derivative of each bin:

$N'_F(t) = -1 * \text{LightNS}(t)$
$E'_F(t) = -1 * \text{LightEW}(t)$
$S'_F(t) = -1 * \text{LightNS}(t)$
$W'_F(t) = -1 * \text{LightEW}(t)$

$N'_L(t) = -1 * \text{LightNS}(t) * (1 + S'_F(t))$
$E'_L(t) = -1 * \text{LightEW}(t) * (1 + W'_F(t))$
$S'_L(t) = -1 * \text{LightNS}(t) * (1 + N'_F(t))$
$W'_L(t) = -1 * \text{LightEW}(t) * (1 + E'_F(t))$

$N'_R(t) = -1 * W'_F(t)$
$E'_R(t) = -1 * S'_F(t)$
$S'_R(t) = -1 * E'_F(t)$
$W'_R(t) = -1 * N'_F(t)$

The definitions above assume constant base changes between all streams, that no new cars are entering the system, and that left turns have zero change whenever the cars coming straight are coming at full capacity. To account for these, we add some constants that allow for configuring the system:

$N'_F(t) = -1 * c_f * \text{LightNS}(t) + add_f$
$E'_F(t) = -1 * c_f * \text{LightEW}(t) + add_f$
$S'_F(t) = -1 * c_f * \text{LightNS}(t) + add_f$
$W'_F(t) = -1 * c_f * \text{LightEW}(t) + add_f$

$N'_L(t) = -1 * c_l * \text{LightNS}(t) * ((left\_turn\_gap\_c * c_f) + S'_F(t)) + add_l$
$E'_L(t) = -1 * c_l * \text{LightEW}(t) * ((left\_turn\_gap\_c * c_f) + W'_F(t)) + add_l$
$S'_L(t) = -1 * c_l * \text{LightNS}(t * ((left\_turn\_gap\_c * c_f) + N'_F(t)) + add_l$
$W'_L(t) = -1 * c_l * \text{LightEW}(t) * ((left\_turn\_gap\_c * c_f) + E'_F(t)) + add_l$

$N'_R(t) = -1 * c_r * W'_F(t) + add_r$
$E'_R(t) = -1 * c_r * S'_F(t) + add_r$
$S'_R(t) = -1 * c_r * E'_F(t) + add_r$
$W'_R(t) = -1 * c_r * N'_F(t) + add_r$

3.2   Results

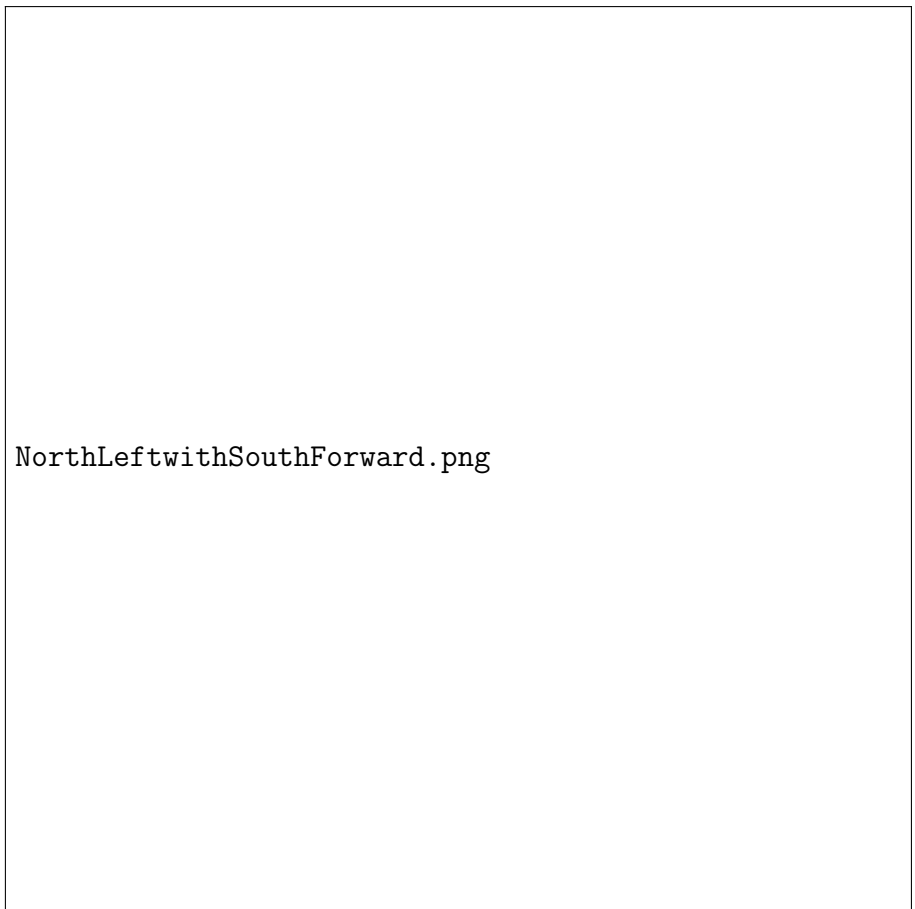Solutions to this system were calculated numerically with scipy.integrate.solve_ivp.

Below we see the North and East forward streams, with the north-south light function shown with the alternating green and red zones. This model aligns with our expectations, with the number of cars waiting to go straight decreasing from the north when the light is green while the east stream accumulates more cars.

Further below, we see the three different north streams. We see that while the light is green the forward and right streams change identically, however, the right stream continues to decrease during the transition from green to red. This again lines up with what we would expect from the rules of the system, as the right turn can continue to turn right up until cars start to flow forward from the east.

Next we examine how a left turn interacts with the forward blocking stream. We see that the left turn has the sharpest decrease right before the light turns red. This is an interesting attribute of the model that lines up with how left turns behave in day to day life. Often, people waiting to turn left will creep into the intersection, then turn left as the light turns red and the oncoming traffic stops.

NorthStreams.png

NorthLeftwithSouthForward.png