Corridor Ramp Metering

Using Particle Filter Model Predictive Control

Vince Wong, Tae Kim, Peilin Li, Daman Heer

TABLE OF CONTENTS

Executive Summary

01

Executive Summary

Background

Over the past 40 years, since the inception of Ramp Metering much research have been conducted for an effective method in order to reduce congestion and improve the freeway operations. To date, none has provided a definitive congestion management tool due to the increased traffic over the same period.

Recommendation

A viable alternative to the current Traffic Responsive Ramp metering methods is proposed that would provide an improved freeway performance, hence a reduced congestion.

Here we propose, an easily implementable path, using the existing equipment to elongate the duration of high traffic volume traffic flow during the high demand hours of operation.

Our Problem

02

Our Problem

+ 54 Hours — On the road
+ $179 Billion — On fuel consumption
+ 167% — Greenhouse gas

Our Problem

The local traffic responsive metering methods is inversely proportional to the density of traffic and is not suited for non linear non gaussian system control resulting in daily congestion experienced throughout all urban areas

Our Problem

How can we embark on Managing our Freeway Traffic more efficiently?

Our Proposal

03

Our Proposal

Particle Filter - Model Predictive Control

We can manage the freeway congestion by using the appropriate ramp metering discharge rates that smooths out the flow parameters thus avoiding the backpropagation of density and limiting drastic reductions in speed. We propose using the most advanced method for the control of non-linear non-Gaussian systems namely ***Particle Filter –Model Predictive Control***

Our Proposal

What is Particle Filter - MPC ?

MPC is a method to use model predictions to plan an ***optimized future trajectory for time-varying systems***. It is often referred to as Model Predictive Control (MPC) or Dynamic Optimization.

Our Proposal

How does it work?

Model predictive control has a number of manipulated variables (MV) and Uncontrolled variables (UCV). The MVs are manipulated numerically to achieve the optimization objective of a desired application performance.

The numerical solution is compared to a desired trajectory and the difference is minimized by re-adjustment of MV in the model so that a response as close as possible to the desired objective is achieved. After first control action is taken and optimized the entire process is repeated at the next time instance (30 seconds). The process is repeated because objective targets may change or updated measurements may require the readjustment of state estimates.

Our Proposal

How does it work?

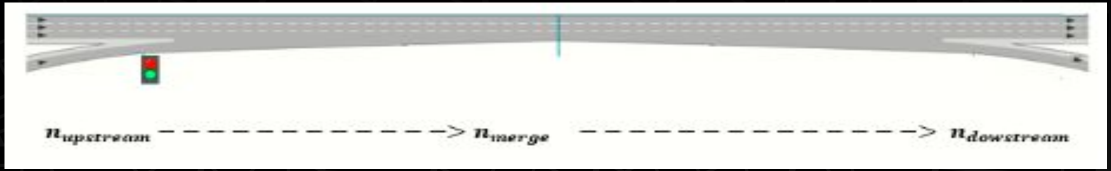
The purpose of the method is to filter out the best manipulated variable MV to accomplish an optimized future trajectory for time-varying systems.

At the start a large number of MVs are tried in the model and the output is compared to a desired trajectory or a measured value. The MV with the best outcome is resampled again to minimize the difference between the outcome and the desired trajectory.

Our Model

04

Density Evolution Model



The density evolution model will follow the transformation of upstream flow parameters of N lane freeway to downstream via the equation:

$$n_{ds}\,(T) = \frac{1}{V_{\;ups}(t)}\left(Q_{ups}\,(t) + \frac{q_{ramp}(t)}{\aleph}\right) - \frac{1}{\aleph}\,\frac{q_{exit}(T)}{V_{\;merge}}$$

Where V merge is given by

$$V_{\;merge} = min\{V_{\;ups}(t),\;\; \frac{5280\,\aleph V_{\;ups}(t)}{\tau\underline{c}(\aleph Q_{ups}(t)+q_{ramp}(t))} - \frac{L}{\tau\underline{c}}\}$$

Density Evolution Model

In this model the Ramp Meter Discharge Rate (qramp) is the manipulated variable MV and the downstream exit rate (qexit) is an uncontrolled variable UCV.

The time T is the delayed response of downstream variables to the upstream actions at the time t.

Our Model Overview

In our model we have applied PF-MPC to control the discharge rates of all ramp meters along a freeway corridor with the goal of smoothing the traffic flow parameters (Volume, Density and Speed) along the freeway to avoid traffic flow breakdown that causes daily traffic congestion.

All discharge rates are tried for every controller along a route at every step of time in the model and the best solution is filtered out to be sent to the controller every 30 seconds

Our Data

In our findings, we demonstrate our conclusions through existing real-time field data from the publicly accessible California Department of Transportation Performance Measurement System (PeMS) and Caltrans District 12 Ramp Metering Information System.

24 hours worth data at 30 seconds interval on Highway -78 was collected from Caltrans RMIS system for the purpose of experimenting with the PF-MPC and training the TensorFlow.

Some freeway sensors had sporadic reporting that was replaced with the previous 30 second data to attain continuity of data.

Tensor Flow

we train a model that takes in the input [upstream volume, upstream speed, discharge rate, number of lanes] to predict the values [downstream volume, downstream speed].
The model that we use is called "Sequential Model". We use TensorFlow to create this model.

We use the data from PeMS to train this model because PeMS data contains all four input and two output data.

Our Results

04

Data from PeMS

	Id	Name	Time	V_ups	Q_ups	q_ramp	q_exit	V_merge	postmile	Q_ds	V_ds
0	201	Jefferson St	05:30:00	73	213	0.0	0.0	73.0	0.858	335.0	63.0
1	200	El Camino Real	05:30:00	63	335	0.0	0.0	63.0	1.594	274.0	59.0
2	199	Plaza Dr	05:30:00	59	274	0.0	120.0	59.0	3.586	335.0	52.0
3	24	Emerald Dr	05:30:00	52	335	0.0	120.0	52.0	4.474	517.0	79.0
4	205	Vista Village Dr	05:30:00	79	517	600.0	120.0	79.0	6.320	548.0	78.0
...	...	...	...	...	...	...	...	...	...	...	...
12594	398	Las Posas Rd/Grand Ave	19:29:00	82	975	0.0	0.0	82.0	11.364	761.0	66.0
12595	180	San Marcos Blvd	19:29:00	66	761	0.0	480.0	66.0	12.274	914.0	68.0
12596	234	Twin Oaks Vlly Rd	19:29:00	68	914	360.0	0.0	68.0	13.022	670.0	67.0
12597	236	Barham/Woodland	19:29:00	67	670	0.0	360.0	67.0	14.860	853.0	62.0
12598	13006	Nordahl Rd	19:29:00	62	853	0.0	0.0	62.0	15.596	792.0	55.0

11760 rows × 11 columns

Input & Output

input

output

	Id	Name	Time	V_ups	Q_ups	q_ramp	q_exit	V_merge	postmile	Q_ds	V_ds
0	201	Jefferson St	05:30:00	73	213	0.0	0.0	73.0	0.858	335.0	63.0
1	200	El Camino Real	05:30:00	63	335	0.0	0.0	63.0	1.594	274.0	59.0
2	199	Plaza Dr	05:30:00	59	274	0.0	120.0	59.0	3.586	335.0	52.0
3	24	Emerald Dr	05:30:00	52	335	0.0	120.0	52.0	4.474	517.0	79.0
4	205	Vista Village Dr	05:30:00	79	517	600.0	120.0	79.0	6.320	548.0	78.0
...	...	...	...	...	...	...	...	...	...	...	...
12594	398	Las Posas Rd/Grand Ave	19:29:00	82	975	0.0	0.0	82.0	11.364	761.0	66.0
12595	180	San Marcos Blvd	19:29:00	66	761	0.0	480.0	66.0	12.274	914.0	68.0
12596	234	Twin Oaks Vlly Rd	19:29:00	68	914	360.0	0.0	68.0	13.022	670.0	67.0
12597	236	Barham/Woodland	19:29:00	67	670	0.0	360.0	67.0	14.860	853.0	62.0
12598	13006	Nordahl Rd	19:29:00	62	853	0.0	0.0	62.0	15.596	792.0	55.0

11760 rows × 11 columns

Training Model

```
# Fit model
def get_compiled_model():
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(16, activation='relu'),
        tf.keras.layers.Dense(32, activation='relu'),
        tf.keras.layers.Dense(2, activation='softmax')
    ])

    tf.keras.optimizers.Adam(
        learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False,
        name='Adam'
    )

    model.compile(optimizer='adam',
                  loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                  metrics=['accuracy']
                 )

    return model

model = get_compiled_model()
model.fit(X_train, y_train, epochs=15)
```

95%

accuracy

Random Discharge Rates

```
q_ramps = [
          1008, 971, 934, 896, 859, 822, 785,
          748, 710, 673, 636, 599, 562, 524,
          484
         ]
```

Optimal q_ramp

	Id	Name	Q_ups	Time	V_ups	q_exit	V_merge	postmile	Q_ds	V_ds	n_ups	q_ramp
1	200	El Camino Real	335	05:30:00	63	0.0	63.0	1.594	274.0	59.0	5.317460	968.0
2	199	Plaza Dr	274	05:30:00	59	120.0	59.0	3.586	335.0	52.0	4.644068	968.0
3	24	Emerald Dr	335	05:30:00	52	120.0	52.0	4.474	517.0	79.0	6.442308	968.0
4	205	Vista Village Dr	517	05:30:00	79	120.0	79.0	6.320	548.0	78.0	6.544304	968.0
5	198	Escondido Ave	548	05:30:00	78	0.0	78.0	6.886	853.0	71.0	7.025641	968.0
...	...	...	...	...	...	...	...	...	...	...	...	...
12594	398	Las Posas Rd/Grand Ave	975	19:29:00	82	0.0	82.0	11.364	761.0	66.0	11.890244	968.0
12595	180	San Marcos Blvd	761	19:29:00	66	480.0	66.0	12.274	914.0	68.0	11.530303	968.0
12596	234	Twin Oaks Vlly Rd	914	19:29:00	68	0.0	68.0	13.022	670.0	67.0	13.441176	968.0
12597	236	Barham/Woodland	670	19:29:00	67	360.0	67.0	14.860	853.0	62.0	10.000000	968.0
12598	13006	Nordahl Rd	853	19:29:00	62	0.0	62.0	15.596	792.0	55.0	13.758065	968.0

Particle Filter

	Id	Name	Q_ups	Time	V_ups	q_exit	V_merge	postmile	Q_ds	V_ds	n_ups	q_ramp
1	200	El Camino Real	335	05:30:00	63	0.0	63.0	1.594	274.0	59.0	5.317460	964.585767
2	199	Plaza Dr	274	05:30:00	59	120.0	59.0	3.586	335.0	52.0	4.644068	964.585767
3	24	Emerald Dr	335	05:30:00	52	120.0	52.0	4.474	517.0	79.0	6.442308	964.585767
4	205	Vista Village Dr	517	05:30:00	79	120.0	79.0	6.320	548.0	78.0	6.544304	964.585767
5	198	Escondido Ave	548	05:30:00	78	0.0	78.0	6.886	853.0	71.0	7.025641	964.585767
...	...	...	...	...	...	...	...	...	...	...	...	...
12594	398	Las Posas Rd/Grand Ave	975	19:29:00	82	0.0	82.0	11.364	761.0	66.0	11.890244	964.585767
12595	180	San Marcos Blvd	761	19:29:00	66	480.0	66.0	12.274	914.0	68.0	11.530303	964.585767
12596	234	Twin Oaks Vlly Rd	914	19:29:00	68	0.0	68.0	13.022	670.0	67.0	13.441176	964.585767
12597	236	Barham/Woodland	670	19:29:00	67	360.0	67.0	14.860	853.0	62.0	10.000000	964.585767
12598	13006	Nordahl Rd	853	19:29:00	62	0.0	62.0	15.596	792.0	55.0	13.758065	964.585767

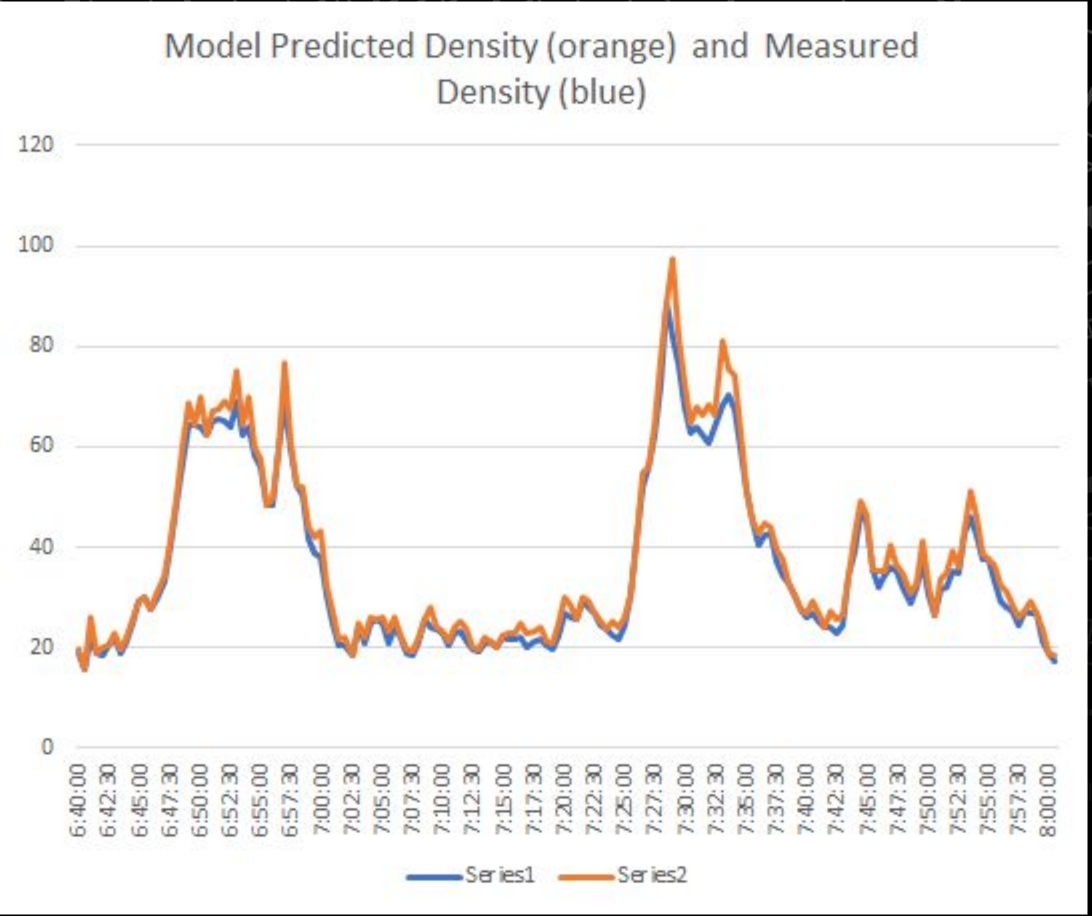
TensorFlow model

```
In [40]:
# split training set
X = input_df[['Q_ups', 'V_ups', 'q_ramp', 'num_lanes']]
y = input_df[['Q_ds', 'V_ds']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)

def get_compiled_model():
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(3, activation='relu'),
        tf.keras.layers.Dense(4, activation='relu'),
        tf.keras.layers.Dense(2, activation='softmax')
    ])

    tf.keras.optimizers.Adam(
        learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False,
        name='Adam'
    )

    model.compile(optimizer='adam',
                  loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                  metrics=['accuracy']
                 )

    return model

model = get_compiled_model()
model.fit(X_train, y_train, epochs=15)
```

```
Epoch 1/15
273/273 [==============================] - 1s 1ms/step - loss: -51.5832 - accuracy: 0.5592
Epoch 2/15
273/273 [==============================] - 0s 1ms/step - loss: -288272.7489 - accuracy: 0.9543
Epoch 3/15
273/273 [==============================] - 0s 973us/step - loss: -2918203.0046 - accuracy: 0.9521
Epoch 4/15
273/273 [==============================] - 0s 1ms/step - loss: -8468978.1825 - accuracy: 0.9587
Epoch 5/15
273/273 [==============================] - 0s 1ms/step - loss: -17393258.7701 - accuracy: 0.9486
Epoch 6/15
273/273 [==============================] - 0s 1ms/step - loss: -30346665.9854 - accuracy: 0.9496
Epoch 7/15
273/273 [==============================] - 0s 1ms/step - loss: -47780554.3650 - accuracy: 0.9495: 0s - loss: -46669
610.8972 - accuracy: 0.
Epoch 8/15
273/273 [==============================] - 0s 1ms/step - loss: -69173429.0073 - accuracy: 0.9515
Epoch 9/15
273/273 [==============================] - 0s 1ms/step - loss: -96371197.7226 - accuracy: 0.9510
Epoch 10/15
273/273 [==============================] - 0s 1ms/step - loss: -127324434.9197 - accuracy: 0.9488
Epoch 11/15
273/273 [==============================] - 0s 980us/step - loss: -168016996.4380 - accuracy: 0.9524
Epoch 12/15
273/273 [==============================] - 0s 1ms/step - loss: -207308297.6934 - accuracy: 0.9512
Epoch 13/15
273/273 [==============================] - 0s 957us/step - loss: -259907161.0511 - accuracy: 0.9525
Epoch 14/15
273/273 [==============================] - 0s 1ms/step - loss: -311698873.4599 - accuracy: 0.9536
Epoch 15/15
273/273 [==============================] - 0s 1ms/step - loss: -372826999.2409 - accuracy: 0.9526

Out[40]: <tensorflow.python.keras.callbacks.History at 0x7fa02c764310>
```

TensorFlow after the training session it was exposed to test data with resulting output at 95% accuracy.

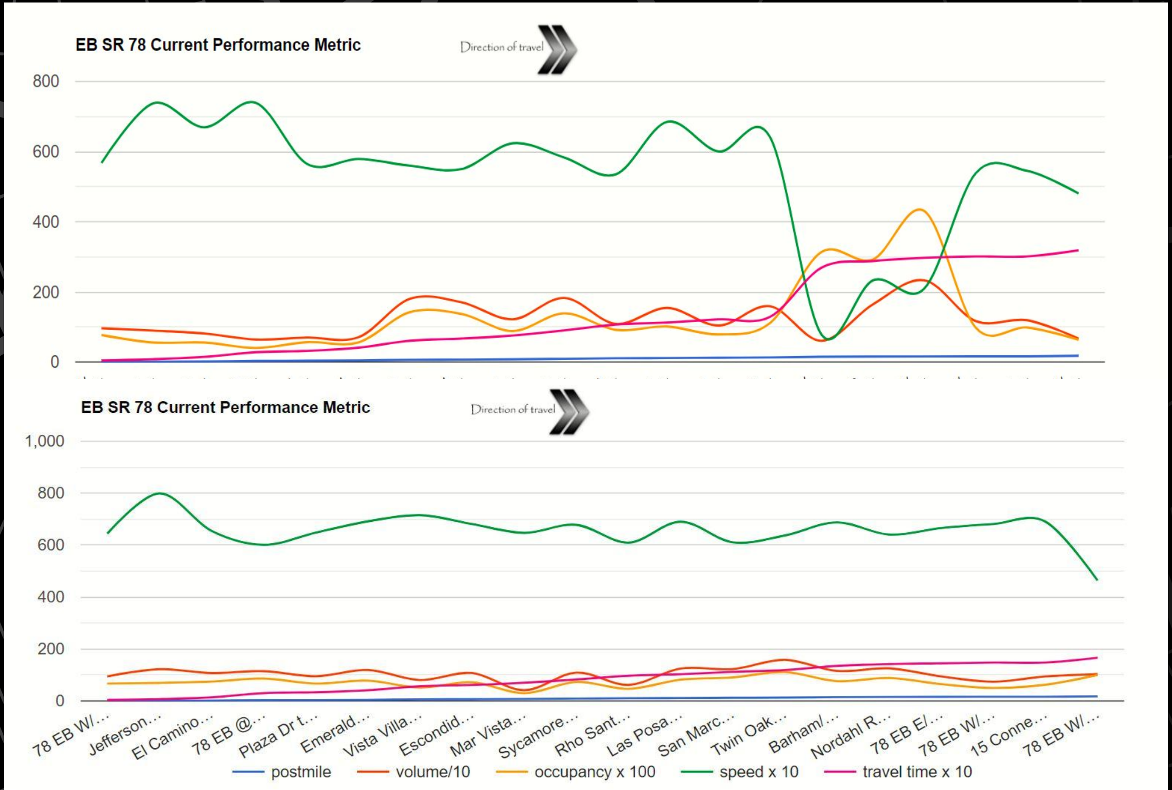
We compared the TensorFlow output for downstream densities to the output density of our model. The TensorFlow output of the downstream density values had close correspondence to the density evolution model



How good is our model?

We tried our traffic model on real time data obtained from Caltrans D11- RMIS system.

The model predictions and the real time data correspond very closely to each other. The slight discrepancies are very minute and can be attributed to the detector’s sensitivity setting in the field.



A snapshot of the Freeway Condition before and after the application of the Particle Filter

Next Steps

Thank you!

To our mentors and teachers
Anush Badii, Natasha Balac, and Leanne Chukoskie

Sources

- D. Schrank, B. Eisele and T. Lomax, "2019 Urban Mobility Report", The Texas A&M Transportation Institute, 2019, pp. 7-8.
- PeMS Performance Measurement System: http://pems.dot.ca.gov
- B. Greenshields, A study of highway capacity. Proceedings of the 14th Annual Meeting Highway Research Board. (PP448-477) 1935. Ohio State Highway Department.
- J. Lighthill, J.B. Whitham, On kinetic waves, II: A theory of traffic flow on long crowded roads. Proc. Royal Society A229 (PP 281-345) 1955.
- P.I. Richards, Shock waves on the highway. Operations Research 4 (PP 42-51) 1956.
- G. F. Newell, A review of traffic flow theory. Berkeley 1965. http://web.pdx.edu/~bertini/courses/559/newell_lecture_65.pdf
- J. H. Banks, Average time gaps in congested freeway flow. Transportation Research, Part A vol. 37, PP 539- 554, 2003
- P. Y. Li, Ankur. Shrivastava, Traffic flow stability Traffic flow stability induced by constant time headway policy for adaptive cruise control vehicles. Transportation Research Part C: Emerging Technologies. 2002;10(4):275-301 Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN.
- Shockwave traffic jams recreated for the first time. Footage courtesy mathematical Society of traffic flow in Japan. https://www.youtube.com/watch?v=Suugn-p5C1M
- [NEW YORK DAILY NEWS, Tuesday, November 5, 2013, 10:39 AM. Congestion cured by cruise control? MIT researcher claims traffic woes could be cut without
- Mathew Bart et al Acces Magazine traffic congestion greenhouse gases https://www.accessmagazine.org/fall-2009/traffic-congestion-greenhouse-gases/