A Matrix-based Quick Programming Tool for Establishing Urban Road Network Topology

User Manual

Yunqing Jia[1], Xiao Chen*

[1] School of Transportation, Southeast University

* College of Transportation Engineering, Chang'an University

JMTS_2202 Project

2023.9.1

Content

1. Introduction
2. Install as a Matlab APP
3. Matlab Source Code
4. Future Works

Content

1. **Introduction**
2. Install as a Matlab APP
3. Matlab Source Code
4. Future Works

Introduction

Motivation

For large-scale urban road networks, this tool aims at generating corresponding matrices that can be applied in establishing road network topology in an efficient approach. The generated matrices can capture upstream-downstream inter-queue relationships, traffic demand, and signal control scheme characteristics.

Contribution

The JMTS_2202 Project was settled to develop such codes (or applications) that can help users to conveniently form up their own urban road network topological matrices. With these matrices, the modeling process of surrogate macroscopic traffic models for large-scale urban road networks (e.g., finite capacity queuing network model) can be partly simplified.

Note: Further details can refer to the JMTS_2202 technical report.

Matlab-based Tool

The 1.0 version of the JMTS_2202 tool is coded based on Matlab language. It provides two application approaches:

- **Install as a Matlab APP**
- **Run Matlab source code**

Both approaches need Matlab software (version: 2022a or later) installed on the user's computer.

For the convenience of the illustration, this manual uses Matlab R2022a installed on a Mac laptop. Detailed information for the example (default) urban road network is in Chapter 4 of the technical report.

Content

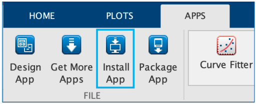
1 Introduction

2 **Install as a Matlab APP**

3 Matlab Source Code

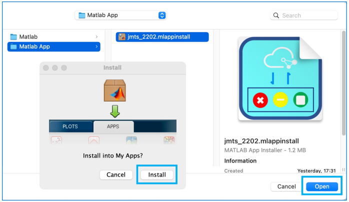
4 Future Works

Install as a Matlab APP

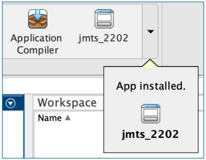


Step 0: Install:

0.1 Open Matlab, Click APPS - Install App.



0.2 Select 'jmts_2202.mlappinstall', then click 'Open' and 'Install'.



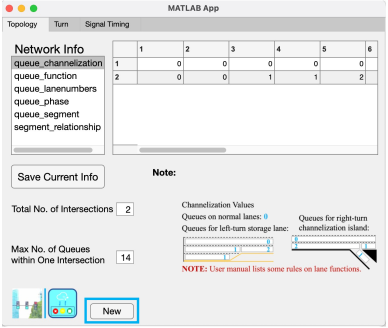
0.3 User can find an app named "jmts_2202" in the Matlab application list.

Step 1: Topology Information:

1.0 Open 'jmts_2202' Matlab APP. The network information of the example network has already been listed.

Create a new folder and set it as the current folder in the Matlab. All new output matrices (new *.mat files) can be found in this folder in the future.

If users would like to edit their own network information, change the value of 'Total No. of Intersections' and 'Max No. of Queues within One Intersection', then click the 'New' button.



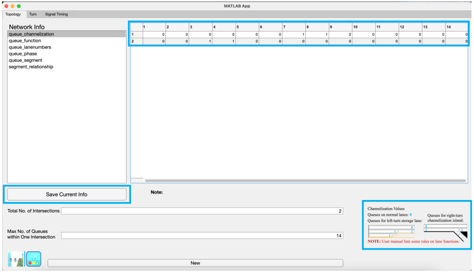
Note: In this tool, the 'Intersections' refers to signalized intersections. Generally, one unsignalized intersection is set between two adjacent signalized intersections. (Section 2.1 of JMTS_2202 technical report)

Step 1: Topology Information: (example)

1.1 Channelization information for queues.

In the channelization information matrix, row *a* represents intersection *a*, and column *b* of row *a* represents b^{th} queue of intersection *a*.

- If queue (a,b) is on normal lanes (or doesn’t exist), matrix (a,b) = 0.
- If queue (a,b) is related to channelization, matrix (a,b) = 1 for downstream queues and = 2 for upstream queues.

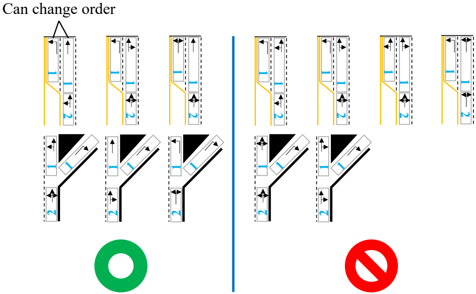


Remember to click the ‘Save Current Info’ button before switching to another info.

Note: One segment can only have one type of channelization!

Step 1: Topology Information: (example)

1.1 Channelization information for queues.
For different scenarios of the channelization, version 1.0 of 'jmts_2202' has the following applicable and inapplicable circumstances.



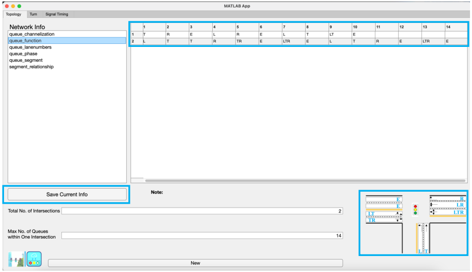
Two downstream queues can change order from each other.
Note: For inapplicable circumstances, if users wish to make them applicable, it is advised to simplify the channelization scheme.

Step 1: Topology Information: (example)

1.2 Function information for queues.

In the function information matrix, row *a* represents intersection *a*, and column *b* of row *a* represents b^{th} queue of intersection *a*.

- If queue (a,b) is on approaching lanes, matrix (a,b) = L, T, R, LT, LR, TR, or LTR according to the real-world function.
- If queue (a,b) is on exit (or receiving) lanes, matrix (a,b) = E.
- If queue (a,b) doesn’t exist, matrix (a,b) is empty.



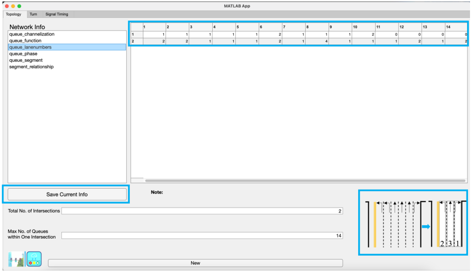
Remember to click the ‘Save Current Info’ button before switching to another info.

Step 1: Topology Information: (example)

1.3 Lane numbers information for queues.

In the function information matrix, row a represents intersection a, and column b of row a represents b^{th} queue of intersection a.

- Users can refer to Section 3.3.1 of JMTS_2202 technical report which demonstrates the reason and the process of determining the number of lanes for each queue.
- If queue (a,b) doesn’t exist, matrix (a,b) = 0.



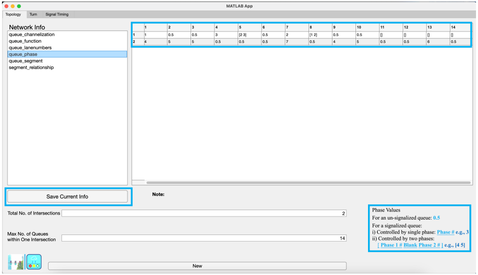
Remember to click the ‘Save Current Info’ button before switching to another info.

Step 1: Topology Information: (example)

1.4 Phase number information for queues.

In the function information matrix, row *a* represents intersection *a*, and column *b* of row *a* represents b^{th} queue of intersection *a*.

- If queue (a,b) is un-signalized, matrix (a,b) $= 0.5$;
- If queue (a,b) is controlled by a single phase numbered P_i, matrix (a,b) $= P_i$.
- If queue (a,b) is controlled by two phases numbered P_{i1} and P_{i2}, matrix (a,b) $= [P_{i1}\ P_{i2}]$.
- If queue (a,b) doesn’t exist, matrix (a,b) $= []$.



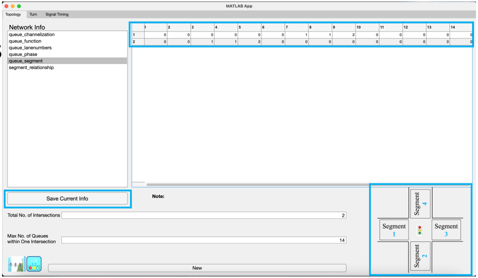
Remember to click the ‘Save Current Info’ button before switching to another info.

Step 1: Topology Information: (example)

1.5 Segment numbers information for queues.

In the function information matrix, row a represents intersection a, and column b of row a represents b^{th} queue of intersection a.

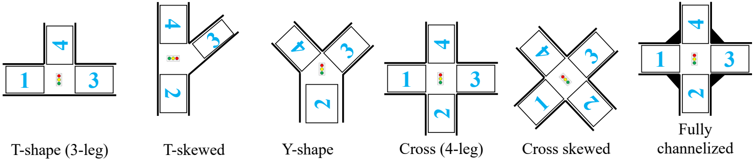
- The setting and numbering rules for determining the segment number for each queue are on the next slide.
- If queue (a,b) doesn’t exist, matrix (a,b) $= 0$.



Remember to click the ‘Save Current Info’ button before switching to another info.

Step 1: Topology Information: (example)

1.5 Segment numbers information for queues.
Version 1.0 of ‘jmts_2202’ accepts the following forms of intersections:



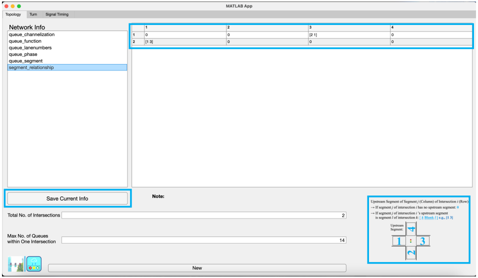
The above figure gives one kind of way to define segment numbers which can be considered as a generalized approach. Users can define their segment numbers in a customized way.
Note: Once the segment numbers are settled, they should remain unchanged throughout the project.

Step 1: Topology Information: (example)

1.6 Segment relationship information.

In the segment relationship information matrix, row *a* represents intersection *a*, and column *b* of row *a* represents the upstream segment of segment *b* of intersection *a*.

- If segment (a,b) has no upstream segment, matrix (a,b) = 0;
- If segment (a,b) 's upstream segment is the segment *d* of intersection *c*, matrix (a,b) = [*c d*].



Remember to click the 'Save Current Info' button before switching to another info.

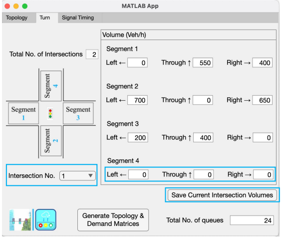
Step 1: Topology Information: (example)

Summary: After step 1, users can generate the following *.mat files:

Name	
rel_seg1.mat	→ Step 1.6
info_queseg.mat	→ Step 1.5
info_quepha.mat	→ Step 1.4
info_quenml.mat	→ Step 1.3
info_quefun.mat	→ Step 1.2
info_quecha.mat	→ Step 1.1

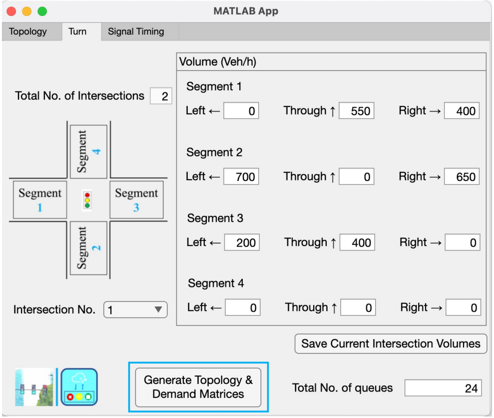
Step 2: Volume Information: (example)
While clicking the 'Turn' tab, this tool can automatically recognize the total number of intersections and the total number of queues based on these files generated from step 1.
For each intersection, please input volume on a segment-directional basis:

- If a segment doesn't exist, all directional volumes of this segment are 0.
- If a segment is a one-way segment without approaching lanes, all directional volumes of this segment are 0.



Remember to click the 'Save Current Intersection Volumes' button before switching to another intersection.

Step 2: Volume Information: (example)
After inputting and saving all volumes of all intersections of the network, click the 'Generate Topology & Demand Matrices' button



Summary: After step 2, users can obtain the TDM.mat file.

‘jmts_2202’ Matlab APP

Step 2: Volume Information: (example)
Summary: In TDM.mat file:

Variable name	Matrix structure	Symbol in technical report
dwnprbind	$I \times B_I$ double	P^B
dwnqueind	$I \times B_I$ double	B^0
extarrind	$I \times 1$ double	Γ^T
info_queind	$I_n \times i_n$ double	-
quenmlind	$I \times 1$ double	${N^I}^T$
totarr	$I_n \times i_n$ double	-
upprbind	$I \times A_I$ double	P^A
upsqueind	$I \times A_I$ double	A^0

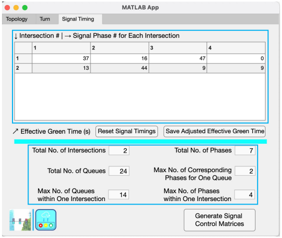
Notations of symbols refer to Chapter 1 of JMTS_2202 technical report.

Step 3: Signal Control Information: (example)

While clicking the 'Signal Timing' tab, this tool can automatically recognize the total number of intersections, the total number of queues, the maximum number of queues within one intersection, the total number of phases, the maximum number of corresponding phases for one queue, and the maximum number of phases within one intersection based on these files generated from step 1.

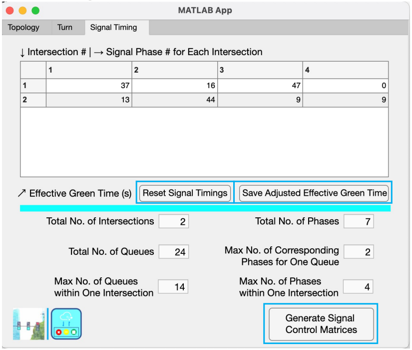
In the signal timing matrix, row *a* represents intersection *a*, and column *b* of row *a* represents the b^{th} signal phase of intersection *a*.

- If a phase doesn't exist, matrix (a,b) = 0.
- If a phase exists, matrix (a,b) = corresponding duration of the signal phase.



‘jmts_2202’ Matlab APP

Step 3: Signal Control Information: (example)
For a new network customized by users, users should click the ‘Reset Signal Timings’ button before filling in the signal timing matrix.
Remember to click the ‘Save Adjusted Effective Green Time’ button after completing the signal timing matrix.



Finally, users can obtain the SCM.mat file by clicking the ‘Generate Signal Control Matrices’ button.

‘jmts_2202’ Matlab APP

Step 3: Signal Control Information: (example)

Summary: In SCM.mat file:

Variable name	Matrix structure	Symbol in technical report
intsgtind	$P_M \times 1$ double	$\mathcal{P}^D$
stsum	$I_n \times P_m$ double	$\bar{\mathcal{P}}^c$
quesrrind	$I \times 1$ double	M^T
TC0	$1 \times I$ double	$\mathcal{C}$
sgm	$I \times S_{max}$ double	$\mathcal{S}^1_{ij}$
shm	$I \times S_{max}$ double	$\bar{\mathcal{S}}^0_{ij}$
stm	$I \times S_{max}$ double	$\bar{\mathcal{S}}^1$
fhm	$I \times S_{max}$ double	$\mathcal{S}^{01}$
stsumh	$I \times P_m$ logical	$\bar{\mathcal{P}}^1$

Notations of symbols refer to Chapter 1 of JMTS_2202 technical report.

Content

1 Introduction

2 Install as a Matlab APP

3 **Matlab Source Code**

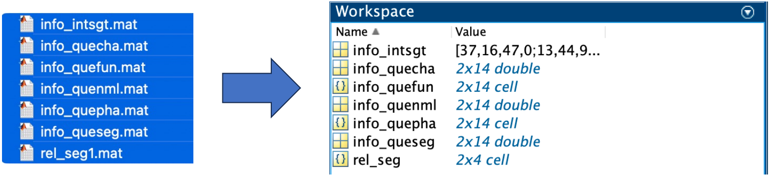
4 Future Works

Run Matlab Source Code

For those users who are familiar with Matlab, they can directly run source codes in the Matlab editor.
Step 0: Edit following variables from *.mat files in the Matlab workspace:

Rules of editing can refer to step 1.1 - step 1.6 and step 3 in the last section.

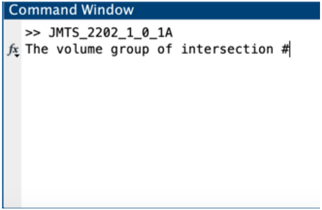
Remember to replace old files with edited *.mat files while keeping the file names unchanged.



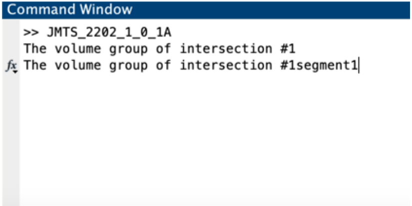
Run Matlab Source Code

Step 1: Run JMTS_2202_1_0_1A.m: (example)

1.1
Input an intersection number in the command window (after the # symbol), then press 'Enter'.



1.2
Input the volume matrix for each segment in the command window (after the # symbol), then press 'Enter'.



Two input matrix formats are as follows: (Veh/h)

- [left-turn-volume through-volume right-turn-volume]
- [left-turn-volume,through-volume,right-turn-volume]

Run Matlab Source Code

Step 1: Run JMTS_2202_1_0_1A.m: (example)

1.3
After finishing all segments in the intersection, press 'Enter' and repeat the step 1.1 and 1.2.

Command Window
```
>> JMTS_2202_1_0_1A
The volume group of intersection #1
The volume group of intersection #1segment1[0,550,400]
The volume group of intersection #1segment2[700 0 650]
The volume group of intersection #1segment3[200 400 0]
The volume group of intersection #1segment4[0 0 0]
fx The volume group of intersection #2
```

1.4
After finishing all intersections, press 'Enter' and users will see '»' symbol in the Matlab command window, which shows the completion of the volume input process.
The volgrp.mat file is generated.

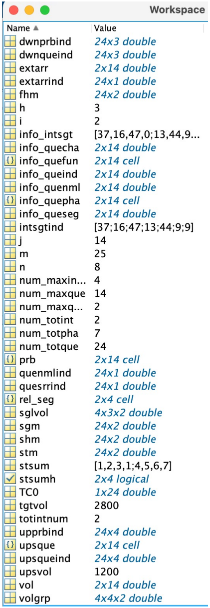
Command Window
```
>> JMTS_2202_1_0_1A
The volume group of intersection #1
The volume group of intersection #1segment1[0,550,400]
The volume group of intersection #1segment2[700 0 650]
The volume group of intersection #1segment3[200 400 0]
The volume group of intersection #1segment4[0 0 0]
The volume group of intersection #2
The volume group of intersection #2segment1[400 2100 300]
The volume group of intersection #2segment2[50 40 30]
The volume group of intersection #2segment3[200 800 100]
The volume group of intersection #2segment4[30 40 50]
fx >>
```

Run Matlab Source Code

Step 2: Run JMTS_2202_1_0_2A.m.

Step 3: Run JMTS_2202_1_0_4A.m.

Finally, all matrices mentioned above (including topology, demand, and signal control matrices) are generated and can be found in the Matlab workspace.


Workspace

Name ▲	Value
dwnprbind	24x3 double
dwnqueind	24x3 double
extarr	2x14 double
extarrind	24x1 double
fhm	24x2 double
h	3
i	2
info_intsgt	[37,16,47,0;13,44,9...
info_quecha	2x14 double
info_quefun	2x14 cell
info_queind	2x14 double
info_quenml	2x14 double
info_quepha	2x14 cell
info_queseg	2x14 double
intsgtind	[37;16;47;13;44;9;9]
j	14
m	25
n	8
num_maxin...	4
num_maxque	14
num_maxq...	2
num_totint	2
num_totpha	7
num_totque	24
prb	2x14 cell
quenmlind	24x1 double
quesrrind	24x1 double
rel_seg	2x4 cell
sglvol	4x3x2 double
sgm	24x2 double
shm	24x2 double
stm	24x2 double
stsum	[1,2,3,1;4,5,6,7]
stsumh	2x4 logical
TC0	1x24 double
tgtvol	2800
totintnum	2
upprbind	24x4 double
upsque	2x14 cell
upsqueind	24x4 double
upsvol	1200
vol	2x14 double
volgrp	4x4x2 double

Content

1 Introduction

2 Install as a Matlab APP

3 Matlab Source Code

4 Future Works

Future Works

1. **This tool would be considered to be reprogrammed via Python.**
2. **Overcome intersection type limitations.**
3. **Overcome one-segment-one-channelization limitation.**
4. **One queue would be controlled by more than 2 phases.**
5. **Overcome inapplicable channelization limitation.**
6. **Optimal signal timings would be calculated automatically.**
7. **Extend more connectors with different kinds of macroscopic traffic models for large-scale traffic signal control.**