SUBPART EIGHT

# Automatic Calibration

CHAPTER 32

# CaDyTS: Calibration of Dynamic Traffic Simulations

Kai Nagel, Michael Zilske and Gunnar Flötteröd

## 32.1 Basic Information

**Entry point to documentation:**

`http://matsim.org/extensions` → cadytsIntegration

**Invoking the module:**

`http://matsim.org/javadoc` → cadytsIntegration → `RunCadyts4CarExample` class

**Selected publications:**

Flötteröd (2010); Flötteröd et al. (2011); Flötteröd et al. (2011a); Flötteröd (2008); Moyo Oliveros (2013)

## 32.2 Introduction

Cadyts (Calibration of Dynamic Traffic Simulations)[1]—licensed under GPLv3 (GNU General Public License version 3.0)—calibrates disaggregate travel demand models of DTA (Dynamic Traffic Assignment) simulators from traffic counts and vehicle re-identification data. Cadyts is broadly compatible with DTA microsimulators, into which it can be hooked through parsimonious interfaces.

As explained formally in Chapter 47 and 48, DTA aims at consistency between a dynamic travel demand model, defining the choice of activity-travel plans, and a dynamic network supply model, capturing spatiotemporal network flows and congestion evolution.

[1] `http://people.kth.se/~gunnarfl/cadyts.html`

**How to cite this book chapter:**

Nagel, K, Zilske, M and Flötteröd, G. 2016. CaDyTS: Calibration of Dynamic Traffic Simulations. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 213–216. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.32. 

Cadyts adjusts the plan choice probabilities of all agents, resulting in simulated network conditions that are consistent with measured real-world data while maintaining the behavioral plausibility of the underlying travel demand model. Within MATSim, plan choice probabilities adjustment is realized by adjusting plan scores, as explained in the next section.

## 32.3 Adjusting Plans Utility

When traffic counts are the empirical source, plan-specific score corrections are composed of link- and time-additive terms $\Delta S_a(k)$ for each link $a$ and each calibration time step $k$ (often one hour). When congestion is light and traffic counts are independently and normally distributed, these correction terms become

$$\Delta S_a(k) = \frac{y_a(k) - q_a(k)}{\sigma_a^2(k)} \tag{32.1}$$

where $y_a(k)$ is the real-world measurement on link $a$ in time step $k$, $q_a(k)$ is its simulated counterpart and $\sigma_a^2(k)$ is (an estimate of) the real measurement variance (assuming its expected value coincides with the prediction $q_a(k)$ of a perfectly calibrated simulator).

The score correction of an agent's given activity-travel plan is calculated as the sum of all $\Delta S_a(k)$, given that following that plan implies entering link $a$ within time step $k$. With this, the *a posteriori* choice probability of agent $n$'s plan $i$ given the count data $\mathbf{y} = \{y_a(k)\}$ becomes

$$P_n(i \mid \mathbf{y}) \sim \exp\left(S_n(i) + \sum_{ak \in i} \Delta S_a(k)\right) = \exp\left(S_n(i) + \sum_{ak \in i} \frac{y_a(k) - q_a(k)}{\sigma_a^2(k)}\right) \tag{32.2}$$

where $S_n(i)$ is the *a priori* score of plan $i$ of agent $n$, as calculated for example with Equation (3.1) and $ak \in i$ reads: "following plan $i$ implies entering link $a$ in time step $k$".

Intuitively, if the simulated value $q_a(k)$ is smaller than the real measurement $y_a(k)$, then a score increase, and thus a choice probability increase, results. The variance $\sigma_a^2(k)$ denotes the level of trust in that specific measurement—a large $\sigma_a^2(k)$ implies a low trust level, taking effect through a large denominator in the corresponding score correction addend.

Flötteröd et al. (2011) is the key methodological reference on Cadyts. It derives the calibration approach from a Bayesian argument and provides more technical information, such as a more general correction of the utility function than in Equation (32.1) that also applies when congestion is present. A lighter presentation is Flötteröd et al. (2011a), where the formulas above are discussed in somewhat greater detail.

## 32.4 Hooking Cadyts into MATSim

Hooking Cadyts into MATSim is based on the following operations:

1. Initialization: When the calibration is started, it requires all available traffic counts and some further parameters. For this, the Cadyts function `void addMeasurement(...)` is called once for every measurement before the simulation starts. It registers a certain measurement type, which has been observed on a specific link.
2. Iterations: The calibration is run jointly with the simulation until (calibrated) stationary conditions are reached.
    a. Demand simulation: The calibration needs an access point in the simulation to affect the plan choice. There are various ways to realize this, depending on the simulator. Before a MATSim agent chooses a plan, it asks the calibration through the Cadyts function

```
double calcLinearPlanEffect(cadyts.demand.Plan<L> plan)
```

for all of this plans' score offsets. The agent then chooses a plan based on accordingly modified scores.
All selected plans of an iteration are registered to Cadyts by

```
void addToDemand(cadyts.demand.Plan<L> plan) .
```

Since Cadyts has its own plans format, MATSim plans need to be converted to that format beforehand.

b. Supply simulation: The calibration must observe simulated network conditions to evaluate their deviation from real traffic counts. For this, the Cadyts function

```
void afterNetworkLoading(SimResults<L> simResults)
```

is called once after each network loading. It passes a container object to the calibration that provides information about the most recent network loading results, particularly on simulated flows at measurement locations.

## 32.5 Applications

Cadyts has been successfully applied in studies like Ziemke et al. (2015); Zilske and Nagel (2015); Flötteröd et al. (2011a). Zürich scenario results illustrate its efficiency, as shown in Flötteröd et al. (2011b, Slide 8), reproduced in Figure 32.1.

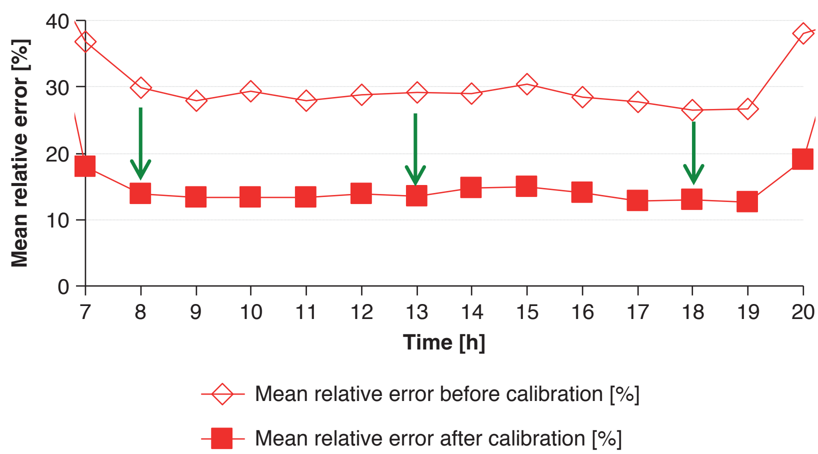


**Figure 32.1:** Zürich case study results: mean relative error in link volumes.
*Source*: Flötteröd et al. (2011b, Slide 8)