Mathematical Institute
Faculty of Mathematics und Physics
Albert-Ludwigs-University Freiburg

# Smart Grid Simulation

**Numerical Optimal Control Project**

in summer semester 2022

by

**Simon Ebert, Simon Sütterlin and Thomas Müller**

**Abstract**

Cost-optimal solutions are highly demanded in volatile sectors for example in the energy sector. We simulate a simple consumer-supply model with one power plant, a group of consumers and a battery for excess power, where we try to optimally control the electricity output of the power plant to minimize costs. The consumption values are modelled with a Markov chain derived from real-world data. We applied a Dynamic Programming algorithm to solve this stochastic optimal control problem and discuss the results and limitations of our approach. The full code to our project is available at `https://github.com/simonsuetterlin/SmartGrid`.

## Contents

**1 Introduction** **1**

**2 Model** **1**

**3 Mathematical Formulation** **3**
3.1 System Dynamics and Loss Function . . . . . . . . . . . . . . . . . . . . . . 4
3.2 Optimal Control Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**4 Optimal Control Algorithm** **5**

**5 Results** **6**

**6 Discussion** **7**

**References** **8**

**Appendix** **8**

# 1 Introduction

The question of optimal power supply has never been as relevant as today. Facing multiple crises like the current energy scarcity connected to the war in Ukraine or the transformation of energy systems to confront global warming, cities search for alternatives in their energy supply. Our goal was to simulate the non-linear interaction between a power plant and a small consumer group to find cost-optimal controls to meet energy demands of the consumer. In the following, we will present the model setup (section 2) and some assumptions (a more detailed list can be found in the Appendix), before explaining the employed optimal control algorithm (section 4). In the end, we will shortly discuss some results and limitations of our model (section 6 and section 5). The full python-code to our project is available at `https://github.com/simonsuetterlin/SmartGrid`.

# 2 Model

We consider a basic power grid with one consumer group, one power plant and one battery. For simplicity, we add an unlimited external energy source as backup supply to eliminate the threat of a power outage. The power plant supplies energy to the consumer and saves excess energy in the battery. The consumer in return tries to compensate his energy needs, first withdrawing from the power plant, second from the battery and all the rest from the external energy source. We assume that excess power which can not be used to charge the battery is transferred to an external power grid for free. That means there are neither costs nor rewards for disposing of energy the power plant has produced superfluously. A scheme of the model setup can be found in Figure 1.

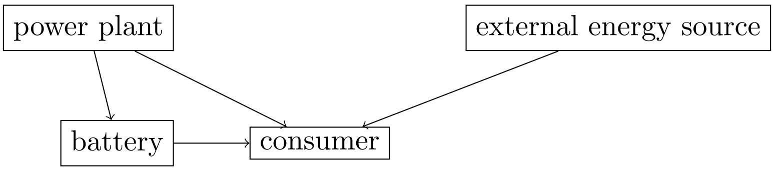


Figure 1: A simple scheme of the model interactions. The arrows symbolize energy flows between the different components

We assume that the power plant has a base price per megawatt hour. When using the energy from the battery, there are additional costs due to charging inefficiencies. The external power, which comes into play only when the battery and the power plant do not meet the consumer's energy demand, is the most expensive. Our optimal control problem is to control the power plant to minimize the costs of covering the electricity demand of the consumer.

The model of our power plant is based on a small industrial gas turbine producing between 10 and 15 MW of power, which is in accordance with current industry standards (Siemens 2022). The consumer is modelled from real-world data (Daignan 2014) from where we obtain a fitting consumer size of around 36 households or a maximum demand of 20 MW, so that production and consumption have roughly the same magnitude. Due to the time interval in the data measurements by Daignan (2014) and the startup time of typical gas turbines (Wärtsilä 2022) we set the time between the controls to 30 minutes. During these intervals, the consumer has piece-wise constant consumption. Their fluctuation is modelled as a Markov chain whose transition matrix is computed from the relative frequencies in the dataset (Daignan (2014), see section 3). The battery in our model can store up to 10 MWh, which is consistent with current battery capacities (Nuclear Power 2022) and has an additional 25% of costs in comparison to the power plant per unit of supplied power. We assume a price of $170 \frac{€}{\text{MWh}}$, which is in the middle range for a peaking power plant with a gas turbine that we are modelling (Lazard 2020) and set the cost for the external power supply to double this value. We are not interested in modelling the distribution of electricity to the consumer and therefore assume that power transfers happen instantly and without energy loss.

We mainly had three hypotheses concerning the behaviour of the model:

(i) In general, the energy production of the power plant follows the consumption pathway.

(ii) Low consumption leads to a complete shutdown of the power plant because having it running at the lowest possible level (which is 10) would be inefficient in this case.

(iii) There are situations where the power plant will be kept running despite lower energy demand in order to recharge the battery.

# 3 Mathematical Formulation

Our state space consists of three dimensions: Power plant output $O$, electricity demand $V$ and battery charge $B$. We can only change the output of the power plant by up to 2 MW per time step or completely shut down or restart the power plant. Therefore, our decision space is one-dimensional. A complete list of the state, decision and random space can be found in Table 1.

| Name | Symbol | Unit | Variable | Definition |
|---|---|---|---|---|
| state-space | $S$ | - | $x$ | $O \times V \times B$ |
| power plant output | $O$ | MW | $o$ | $\{0, 10, 11, 12, 13, 14, 15\}$ |
| electricity demand | $V$ | MW | $v$ | $\{0, \ldots, 20\}$ |
| battery charge | $B$ | MWh | $b$ | $\{0, \ldots, 10\}$ |
| decision space | $U$ | $\delta$MW | $u$ | $\{\text{off}_0, -2, -1, 0, 1, 2, \text{on}_{12}\}$ |
| random space | $\Omega$ | - | $\omega$ | - |

Table 1: Spaces

Since the consumption of households is not entirely predictable, we need to represent the randomness in some manner. Therefore, we calculated the transition matrix of a Markov chain from 25000 real data samples from Daignan (2014). An illustration of the probabilities can be found in Figure 2. The exact values are displayed in Figure 6 in the Appendix. Thus, we obtain the probabilities of the random variable $\omega$ for the consumption $V$, which will be used in section 4.

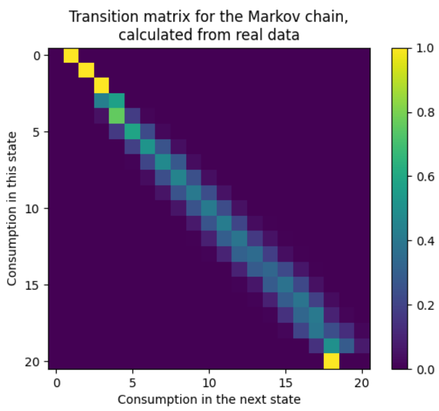


Figure 2: An illustration of the transition matrix for the Markov process, calculated from the real data in Daignan (2014). As usual, the matrix is a right stochastic matrix i.e. the probability of going from state $i$ to state $j$ $\mathbb{P}(j|i)$ is the entry $P_{i,j}$. Note that the probabilities have weak data support in the extremes because the lowest and highest consumption values occur only very rarely in the real world data.

## 3.1 System Dynamics and Loss Function

We define a Markov Step as a function that maps a state $v$ to another state $\tilde{v}$. $\tilde{v}$ is a draw from $V$ based on the transition probabilities for $v$.

$$\begin{aligned} f: \quad S \times U \times \Omega \quad &\to S \\ (x, u, \omega) \quad &\mapsto x' = (o', v', b') \end{aligned}$$

$$o' = O^f(x, u), \quad v' = M_v^f(\omega), \quad b' = B^f(x, o', v')^t$$

with

$$\begin{aligned} O^f: \quad & S \times U && \to O \\ & (x, u) && \mapsto \begin{cases} 0 & \text{if } u = \text{off}_0, \\ 12 & \text{if } u = \text{on}_{12}, \\ \max\{\min\{o + u, 15\}, 10\} & \text{else.} \end{cases} \\ M^f: \quad & \Omega && \to \{V \to V\} \\ & \omega && \mapsto \left\{ v \mapsto M_v^f(\omega) \right\} \\ B^f: \quad & S \times O \times V && \to B \\ & (x, o', v') && \mapsto \max\left\{ \min\left\{ b + \tfrac{1}{2} \cdot (o' - v'), 10 \right\}, 0 \right\}. \end{aligned}$$

The loss function we look at is given by the costs of the power from all supplies, as

$$\begin{aligned} L: \quad S \times S \quad &\to \quad \mathbb{R} \\ x, x' \quad &\mapsto \quad L_i(x, x') + L_e(x, x') + L_b(x, x'). \end{aligned}$$

Each cost function is defined as

$$\begin{aligned} L_i(x, x') &= \text{produce}_O(o') \cdot P_i, \\ L_b(x, x') &= \text{batteryUsed}(o', v', b) \cdot P_b, \\ L_e(x, x') &= (\text{deficit}_O(o', v') - \text{batteryUsed}(o', v', b)) \cdot P_e. \end{aligned}$$

where produce$(\cdot)$ gives the megawatt hour production of the power plant, batteryUsed$(\cdot)$ gives the energy consumption from the battery and deficit$(\cdot)$ gives the total deficit of energy from the power plant versus the consumer. Hence, the amount of imported energy is the difference of our deficit and the amount of battery used.

The terminal costs are given by the costs to completely charge the battery

$$E(x) = (10 - b) \cdot P_i.$$

## 3.2 Optimal Control Problem

Let $x_0$ be the initial state, $\omega = (\omega_0, \ldots, \omega_{T-1}) \in \Omega^T$ our stochasticity and $u = (u_0, \ldots, u_{T-1})$ our decisions. Then

$$x_k \quad = \quad f(x_{k-1}, u_{k-1}, \omega_{k-1}) \quad \text{for all } k = 1, \ldots, T$$

gives the sequence of states we attain with these decisions and this stochasticity.

The information $\mathcal{I}_k$ we use to pick the control $u_k$ is just our current state $x_k$, so we just call them $x_k$. We define policy functions as $\pi = (\overline{u}_0, \pi_1(\cdot), \dots, \pi_{T-1}(\cdot))$, where $\pi_k : S \longrightarrow U$ for all $k = 1, \dots, T-1$. For a given policy $\pi$ and a given initial state $x_0$ we denote the states as $x_k^\pi(x_0)$ and the decisions as $u_k^\pi(x_0)$, more precise we define

$$\begin{array}{rcll} u_0^\pi & = & \overline{u}_0 & \\ u_k^\pi(x_0) & = & \pi_k(x_k) & \qquad \forall k = 1, \dots, T-1. \\ x_k^\pi(x_0) & = & f(x_{k-1}, u_{k-1}^\pi, \omega_{k-1}) & \end{array}$$

Our goal is to find the policy $\pi$, which attains the following minimum:

$$\min_{\pi(\cdot)} \mathbb{E}_{\Omega^T} \left[ \sum_{k=0}^{T-1} L\left(x_k^\pi(x_0), x_{k+1}^\pi(x_0)\right) + E(x_T(x_0)) \right] \tag{1}$$

# 4 Optimal Control Algorithm

We use a Dynamic Programming approach to solve our optimal control problem. We start by determining the terminal costs $E(x)$ of every state $x$ and subsequently doing a backward sweep to obtain the costs-to-go $J_k(x)$ of every state by the following algorithm:

$$\begin{aligned} J_T(x) &= E(x) \\ J_k(x) &= \min_{u \in U} \mathbb{E}_\omega \left[ L(x, f(x, u, \omega)) + J_{k+1}(f(x, u, \omega)) \right] \end{aligned} \tag{2}$$

It is important to note that we have to include the expected value in this equation to account for the randomness $\omega$ of our consumption variable $v$. In our case increasing the simulation depth above five shows no further improvement and therefore we only calculate the cost-to-go to a depth of five. We save the final cost-to-go-values $J_0(x)$ for every state $x$. Concisely speaking, we calculate all the values of the function

$$\begin{array}{rrcl} J_0 : & O \times B \times V & \longrightarrow & \mathbb{R} \\ & \underbrace{(o, b, v)}_{=:x} & \mapsto & \text{Cost-to-go of state } x \end{array}$$

Because we have a three-dimensional state space (power-plant output, battery charge and consumption) we can save these values into a vector-valued matrix (or 3D matrix). For the consumption states 5 and 15 the cost-to-go matrices can be found in Figure 3. There, one can observe, that a higher charge of the battery implies a lower cost-to-go and a higher consumption leads to a higher cost-to-go. However, a higher output does not always mean lower cost-to-go, as the left graphic shows: Here the lag of the power plant (which can only be controlled in steps of 2 MW) comes into play, which is why the state $(o, v, b) = (13, 5, 7)$ has lower costs-to-go than the state $(o, v, b) = (14, 5, 7)$.

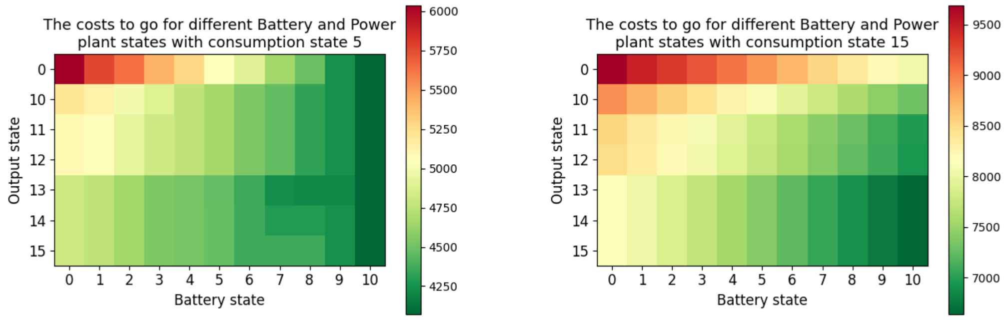


Figure 3: The expected costs-to-go for different values of the output and battery charge for the fixed consumption values 5 and 15. The unit for the costs is €.

With the help of the cost-to-go-functions, one can calculate the optimal decision $u^*$ for the current state $x$ by employing the equation

$$u^*(x) = \underset{u \in U}{\operatorname{argmin}} \, \mathbb{E}_\omega \left[ L(x, f(x, u, \omega)) + J_0 \left( f(x, u, \omega) \right) \right] \tag{3}$$

Due to the fact that our model is time-invariant, we can use the already calculated cost-to-go-matrix to obtain the optimal decision for the next state. Therefore, once the cost-to-go-matrix is calculated, we only have to evaluate Equation 3 in each state to find the optimal decision and hence, simulating the model after $J_0$ was calculated is very efficient.

## 5 Results

To show how our optimal control algorithm performs, we let it run with the real-world data sets we already used in section 2.

The pathway of the energy production by the power plant confirms in a large part the expectations from section 2. One example of such a pathway obtained from real-world data over 4 days, can be found in Figure 4. It shows the day-night-cycle in the real-world data and the emerging shutdown of the power plant during the low-demand night hours.

Additionally, one can see the influence of the battery charge on the power plant control. For example, in the beginning, energy production surpasses consumer demand significantly so that the battery is being charged. One more example of a simulation using real-world data can be found in the Appendix in Figure 5. For comparison to the real-world data, one can find a simulation with a consumer realisation that was simulated from our Markov chain (see section 2). Due to the characteristics of a Markov chain, this last simulation does not include a day-night cycle.

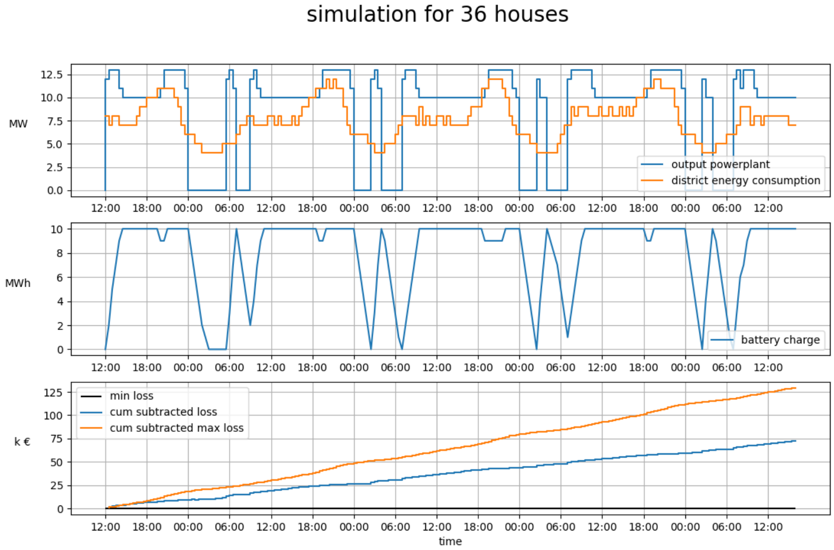


Figure 4: Results of one simulation of the optimal control algorithm with real-world data. The first graph shows the progression of the energy production of the power plant and the energy consumption respectively. The second graph displays the state of the battery charge. In the third graph, one can see the losses for three different scenarios: The orange line shows the loss if all the power was supplied from the external energy source and the black line shows the loss if the power plant supply would always meet the energy demand perfectly. The blue line represents the loss which occurred from our optimization. The loss values were re-scaled by subtracting the minimum loss (black line).

# 6 Discussion

Of course, our model is based on many different simplifications and assumptions (see section 6 for a detailed list). The discretization of time and state spaces leads to less precise results. Furthermore, one could contest our choice of constants (like the power prices for example), which we based on concrete data, but which are nonetheless arbitrary to a certain degree. Hence, the model is not suited for actually controlling the power supply of a minor city, but should rather be seen as a prototype approach to how to deal with the problem.

One concrete improvement of the optimal control algorithm could be to further specify the random process by including more data dimensions, like the time of the day for example. Including this modification would need a lot more computational power and would reduce the number of data samples for each state, but could significantly improve the results.

Despite these simplifications, the model reacts well to random changes in the consumption and reduces losses significantly, while being computation-wise efficient enough to run in approximately two minutes on a Laptop.

## References


Daignan, J.-M. (2014). *Smart meters in London.* URL: https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london.

Lazard (Oct. 19, 2020). "Lazard's Levelized Cost of Energy Analysis, Version 14.0". In: URL: https://www.lazard.com/perspective/lcoe2020 (visited on 08/02/2022).

Nuclear Power (2022). *Electrochemical Energy Storage – Battery Storage Power Station.* URL: https://www.nuclear-power.com/energy-conversion-and-storage-efficiency/electrochemical-energy-storage-battery-storage-power-station/ (visited on 08/02/2022).

Siemens (2022). *SGT-400. Industrial gas turbine.* URL: https://www.siemens-energy.com/global/en/offerings/power-generation/gas-turbines/sgt-400.html.

Wärtsilä (2022). *Technical comparisons: Combustion engines and gas turbines.* URL: https://www.wartsila.com/energy/learn-more/technical-comparisons/combustion-engine-vs-gas-turbine-startup-time (visited on 08/02/2022).


## Appendix

### Assumptions:

(i) All costs are constant in time.

(ii) All variable realisations have to be in the given state spaces.

(iii) All variables (consumption, production, battery charge) are piece-wise constant in time.

(iv) There is no energy loss during power transfers.

(v) The power plant can be modified in steps of up to 2 MW, shutdown completely or restarted from 0 to an output of 12 MW.

(vi) The modeling time step is 30 min.

(vii) The real-world data has a Markovian distribution.

(viii) The external energy source is infinite and always available.

(ix) Excess electricity can be transferred to an external power grid for free.

(x) Charging inefficiencies of the battery are modeled as a higher energy price of the battery. However, the battery does not discharge over time.

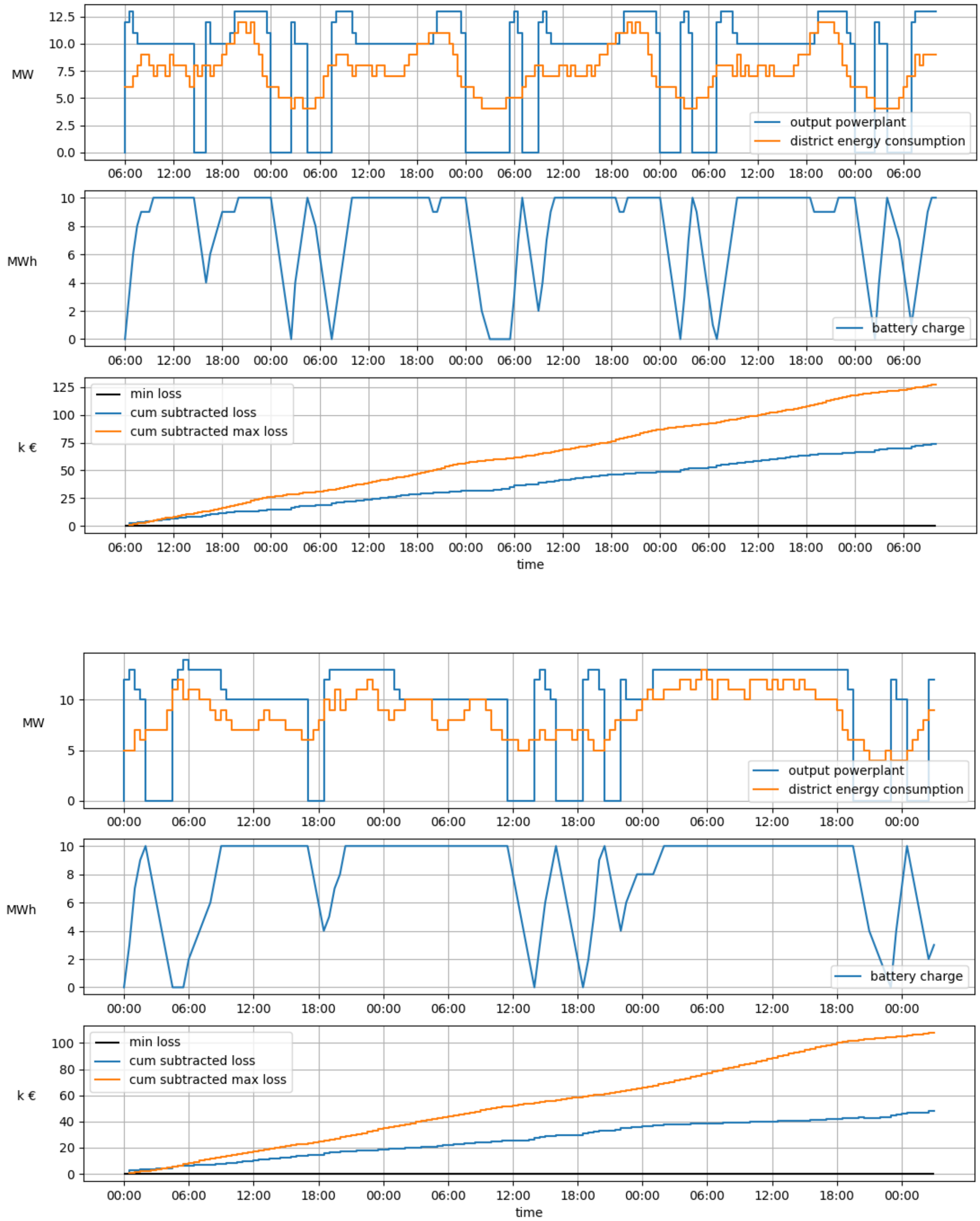


Figure 5: Results of one simulation of the optimal control algorithm with a real-world data set (top) and a simulation of the Markov chain (bottom). For more information see Figure 4

$$
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.430 & 0.570 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.047 & 0.762 & 0.179 & 0.012 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.167 & 0.583 & 0.226 & 0.023 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.002 & 0.188 & 0.516 & 0.254 & 0.038 & 0.002 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.011 & 0.208 & 0.473 & 0.276 & 0.032 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.032 & 0.239 & 0.449 & 0.243 & 0.035 & 0.002 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.052 & 0.256 & 0.402 & 0.238 & 0.049 & 0.002 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.003 & 0.056 & 0.255 & 0.410 & 0.231 & 0.043 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.004 & 0.070 & 0.260 & 0.420 & 0.208 & 0.034 & 0.005 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.006 & 0.086 & 0.299 & 0.385 & 0.168 & 0.050 & 0.006 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.010 & 0.098 & 0.309 & 0.351 & 0.184 & 0.039 & 0.007 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002 & 0.096 & 0.285 & 0.332 & 0.220 & 0.059 & 0.006 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.010 & 0.062 & 0.287 & 0.376 & 0.218 & 0.045 & 0.002 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.010 & 0.092 & 0.291 & 0.386 & 0.189 & 0.031 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.010 & 0.141 & 0.332 & 0.405 & 0.098 & 0.015 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.019 & 0.204 & 0.389 & 0.241 & 0.130 & 0.019 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.143 & 0.500 & 0.286 & 0.071 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
$$

Figure 6: The transition matrix for the Markov process, calculated from the real-world data in Daignan (2014) as described in section 2. As usual, the matrix is a right stochastic matrix i.e. the probability of going from state $i$ to state $j$ $\mathbb{P}(j|i)$ is the entry $P_{i,j}$. Note that the probabilities have weak data support in the extremes because the lowest and highest consumption values occur only very rarely in the real-world data.