Ref. Ares(2022)1490238 - 28/02/2022

H2020 – LC – SC3 – EE – 2019 – GA 894240

Operating System for Smart Services in Buildings

D3.3 Flex-Offer Extensions for Predictions, Flexibility, and Uncertainty Modelling

WP3 Common Ontology and Semantics

	Name	Date
Prepared by	Fabio Lilliu (AAU), Amir Laadhar (AAU), Torben Bach Pedersen (AAU), Christian Thomsen (AAU)	19.01.2022
Peer reviewed by	Yves Stauffer (CSEM), Dominique Gabioud (HES-SO)	14.02.2022
Reviewed and approved by	Dominique Gabioud (HES-SO)	24.02.2022

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 894240.

Distribution list

External		Internal	
European Commission	x	Consortium partners	x

Change log

Version	Date	Remark / changes
1.1	19.01.2022	First version by AAU
2.1	07.02.2022	Second version by AAU
3.1	24.02.2022	Final version by HES-SO

To be cited as

"D3.3 Flex-Offer Extensions for Predictions, Flexibility, and Uncertainty Modelling" of the HORIZON 2020 project domOS, EC Grant Agreement No 894240.

Disclaimer

This document's contents are not intended to replace the consultation of any applicable legal sources or the necessary advice of a legal expert, where appropriate. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user, therefore, uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

Table of contents

1. Contents

List of figures

List of tables

Terms, definitions, and abbreviated terms

GA	Grant Agreement
ICT	Information and Communication Technology
dCO	domOS Common Ontology
FO	FlexOffer
SFO	Standard FlexOffer
TEC	Total Energy Constraint
DFO	Dependency FlexOffer
UFO	Uncertain FlexOffer
LTI	Linear Time Invariant
PDF	Probability Distribution Function

Executive Summary

The purpose of this document is the description of the work carried out in Task 3.3 about FlexOffers (FOs). The main purpose of this task is to extend FOs so that they consider predictions, flexibility, and uncertainty for measures different from energy. First, a description of FOs will be provided; after that, it will be described how FOs can be extended in the way intended for the task. Finally, there will be a short demonstration on how those extended FOs work, what results can be achieved from their exploitation and the code that has been used for their implementation.

1. Introduction

The objective of D3.3 is to describe the extension of FOs to incorporate predictions, flexibility, and uncertainty for several types of measures. This document is organized as follows:

- Chapter 2 (FlexOffers): gives an introduction on FOs, and how they model flexibility.
- Chapter 3 (Extensions of FlexOffers for uncertainty): describes how FOs have been extended in order to incorporate uncertainty.
- Chapter 4 (Extensions for other measures, and prediction modelling): shows how FO can be used to model several types of measures, and how they can be used to model predictions.
- Chapter 5 (Demonstration): describes how certain types of extended FOs can be generated and shows a use case for their exploitation.
- Chapter 6 (Conclusion): concludes the work done in this deliverable and describes future work for this WP.

This deliverable has two main purposes: to show how FOs can model uncertainty (Chapter 3), and to show how FOs can be used to capture measures different from energy (Chapter 4). Chapter 3 will lead to the definition of a new type of FO, called uncertain FlexOffer (UFO), which allow the FO model to capture uncertainty. In chapter 4, we use FOs to model non-energy measures: in section 4.1 we use standard FOs and do not consider uncertainty, while in section 4.2 we model uncertainty for those measures by using UFOs.

2. FlexOffers

This chapter will describe FOs, their purpose, their life cycle, and their functioning on a conceptual level.

2.1. Introduction

In the last decades, the use of Renewable Energy Sources (RES) is becoming more prominent in electrical grids. The way they generate energy is not controllable in many cases, such as wind and solar production, and it depends on variables such as time of the day and the year, or weather. The capability of adjusting energy demand to RES production is therefore very valuable; to this purpose, some grid users (called *prosumers*) are able to change their energy consumption in time and amount. This ability is called *flexibility*. In domOS, an ecosystem for smart services in buildings is developed, and flexibility is one of these services.

Description of flexibility is a concept that has been treated extensively in the literature. Various mathematical models have been created, with different properties depending on the considered cases. The FO model was created in order to be capable to *capture* flexibility from many different types of devices, *optimize* the flexibility for generic purposes (e.g., reducing energy costs, peak shaving), *aggregate* the flexibility from many small energy loads into a few bigger ones, and do the opposite process (*disaggregation*) in order to control the actual loads. Therefore, FOs have the following properties**:** i) they can model flexibility from different device types in a unified format; ii) they can capture most/all of the total flexibility that is available from each device; iii) they are scalable with respect to optimization for long time horizons, and aggregation of many loads. However, there is a fourth aspect that needs to be treated: flexibility in buildings, which is at the core of the service deployed in domOS, is subject to uncertainty. Therefore, it would be important to have a model which complies to the previously descripted properties, and to the following: iv) the model should be able to consider uncertainty related to flexibility. Many models in the literature focused on one or more of these aspects, although none has been considering all of them at the same time, and in particular uncertainty. For representing flexibility from different devices in an unified format, whose importance is discussed in (Junker, et al., 2018), the models from (Schott, et al., 2019) and (Corsetti, et al., 2021) provide some examples. Regarding accuracy, linear time invariant (LTI) state-space models (Borrelli, et al., 2017) (Koller, et al.) are very precise in capturing flexibility for batteries and heat pumps, and the one from (Junker, et al., 2020) can accurately represent flexibility for building heating systems and water towers. In particular, the FlexOffer (FO) (Pedersen, et al., 2018) is a model which generates good approximations of flexibility for many different types of loads , which can be aggregated and optimized in a scalable way (Siksnys, et al., 2016), thus effectively addressing properties i) – iii). However, property iv) has not been considered yet: the work in this deliverable will address this point, while still retaining properties i) – iii).

2.2. Running Example

We will now describe a specific energy load, which will be used through this deliverable as an example for describing the concepts that will be introduced. We will consider a *Tesla Powerwall* battery. Its capacity is 14 kWh, its maximum charging and discharging power are both 5 kW, and its round-trip efficiency is 90%. We use one hour time units, i.e., the battery can either be charged or discharged by an amount up to 5 kWh at each time unit.

For describing the functioning of the battery, we use Coulomb counting (Meng, et al., 2019). At each time unit t, we write the state of charge (SoC) of the battery as:

$$SoC(t) = SoC(t-1) + L \cdot u^{+}(t) + L^{-1} \cdot u^{-}(t)$$

$$SoC_{min} \leq SoC(t) \leq SoC_{max} \qquad \frac{E_{min}}{L} \leq u(t) \leq E_{max}$$

Here:

- $SoC(t)$ is the amount of energy in the battery at time t, expressed in kWh.
- $u(t)$ is the amount of energy that the prosumer gives to/receives from the battery at time t, in kWh: $u(t)$ is positive if the battery is being charged, negative otherwise. $u^{+}(t)$ is max {$u(t)$,0}, $u^{-}(t)$ is min {$u(t)$,0}.
- L is a real number that measures how much energy is kept while charging/discharging the battery: it goes from 0 (all the energy is lost) to 1 (no energy is lost).
- SoC_{min} and SoC_{max} are the minimum and maximum state of charge that the battery can have in kWh, respectively.
- E_{min} and E_{max} are the minimum and maximum amount of energy (in kWh) that can be taken from/given to the battery in one time unit. In the example we are considering: SoC_{min} = 0 kWh, SoC_{max} = 14 kWh, E_{min} = -5 kWh, E_{max} = 5 kWh, L ~ 0.948. There are two use cases that we are considering: in the first, $SoC(1)$ = 0 kWh and the battery can only be charged, in the second, $SoC(1)$ = 7 kWh and the battery can be either charged or discharged at each time unit. We refer to these as the *charging* and *switching* cases, respectively.

2.3. FlexOffer Life Cycle

The baseline for this work is the FO model (Ferreira, et al., 2014) (Pedersen, et al., 2018), already mentioned in Section 2.1. Suppose we want to model flexibility for a certain device: a FO can be seen as a set of constraints on the values of the consumable energy for the following time units, which describe the flexibility available from said device. There are many techniques for generating FOs, depending on the type of device and the type of approximation needed.

We will now describe the life cycle of an FO, shown in Figure 1. Two main parties are involved. The first is the prosumer, who generates and executes the FO; the second processes and issues schedules for the FO, and many different energy market actors can take this role. In (Neupane, et al.), this task belongs to the aggregator, so we will consider the aggregator as the processing party.

Reading Figure 1 from left to right, the prosumer starts by forecasting flexibility for his/her devices and generating FOs according to that. After this, FOs are sent to the aggregator, which will determine if the FO is useful for its needs; after that, it decides whether to accept the FO or not and informs the prosumer of the response. If the FO is not accepted, it is not executed, and the cycle ends here. If the FO is accepted, the aggregator processes it (e.g., aggregating it with other FOs, performing optimization), and establishes a schedule for the FO, eventually after performing disaggregation. The FO schedule is then sent back to the prosumer, which will then execute it by controlling the device.

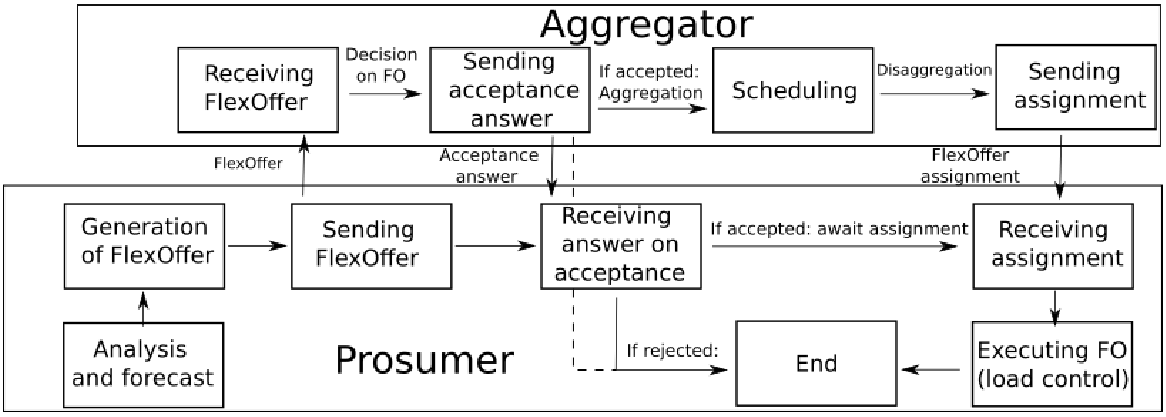

FIGURE 1: FLEXOFFER LIFE CYCLE

2.2 Types of Constraints

An FO is an approximation of the available flexibility, expressed in terms of constraints over the usable amount of energy at each time unit. In this section, call e_t the amount of energy consumed at time t, and T the time horizon we are considering.

There are many types of constraints that have been used to define FOs. The simplest ones are called *slice constraints*, which are divided into *start time constraints* and *energy constraints*. A start time constraint determines the earliest and latest time unit at which the load can start. An energy constraint establishes, for each time unit at which the load is operating, the minimum and maximum amount of energy that can be consumed from that load. This means that for every time unit t, the energy constraint specifies a lower and an upper bound $emin_t$ and $emax_t$ such that $emin_t \le e_t \le emax_t$. A *standard FO* (SFO) is an FO whose constraints are all slice constraints.

In more formal terms, a standard FO is defined as a tuple$(t_{es}, t_{ls}, [emin_1, emax_1], \ldots, [emin_T, emax_T])$. t_{es} and t_{ls} indicate respectively the earliest and latest starting time for the load, T is the duration (in time units) of the device operation once the device is activated, and for every $\mathrm{k} \in \{1, \ldots, \mathrm{T}\}$, $emin_k$ and $emax_k$ are respectively the minimum and maximum consumable amount of energy from the device at time unit k after the activation.

Another type of constraint is the *total energy constraint* (TEC), which specifies the lower (TE_{min}) and upper (TE_{max}) bounds for the energy that can be consumed over the considered time horizon. With the notation used before, this means:

$$TE_{min} \le \sum_{t=1}^{T} e_t \le TE_{max}$$

A *total energy constraint standard FO* (TEC-SFO) is an FO with slice and total energy constraints.

A further type of constraint is the *dependent energy constraint*. This constraint specifies at each time unit t a lower and an upper bound on the amount of energy that can be consumed, depending on the total amount of energy that has been consumed before time unit t. In more formal terms, this means that there are three real numbers a, b, c such that:

$$a \cdot (e_1 + \cdots + e_{t-1}) + b \cdot e_t \leq c.$$

A *dependency FO* (DFO) is an FO with dependency energy constraints.

An FO can approximate flexibility in two main ways: *inner* and *outer*. In an inner approximation, the amount of modelled flexibility is less than the actually available amount: this means that there may be some flexibility not modelled by the FO. In an outer approximation, the modelled flexibility is more than the actually available flexibility: this means that the FO models all the available flexibility, but also models some flexibility that is not actually available. Outer approximation FOs can therefore generate more flexibility compared to inner FOs, but some of the modelled configurations may actually be unfeasible.

Figure 2 shows the constraints described in this section for the *charging* case of the running example. In particular, (a) represents an inner approximation SFO, and (b) an outer approximation SFO: for each time unit the column describes how much energy can be used for charging the battery, and the horizontal line shows an example schedule. A TEC-SFO is represented in (c), with the TEC shown above the slice constraints, and (d) shows a DFO: for each time unit, the x axis indicates how much energy as been consumed in total before time *t*, and the y axis indicates how much energy can be consumed based on the value on the x axis.

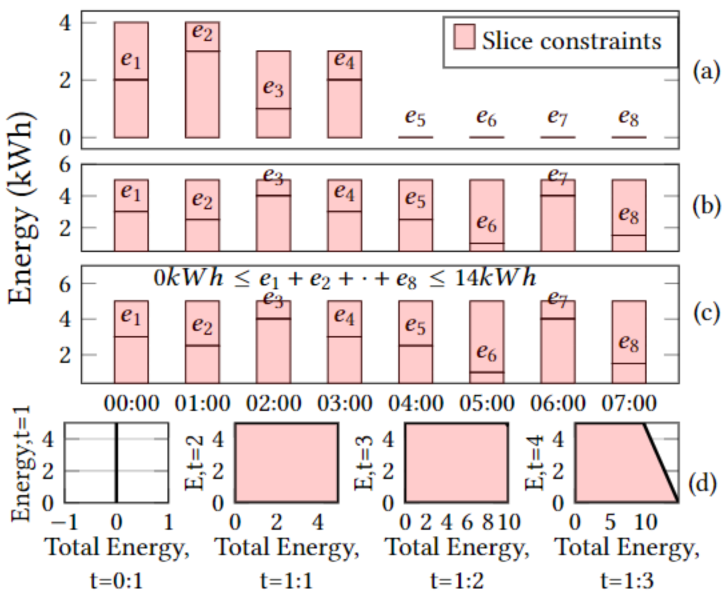

FIGURE 2: INNER (A) AND OUTER (B) SFOS, TEC-SFO (C) AND DFO (D)

3. Extension of FlexOffers for Uncertainty

3.1. Uncertainty Definition

Until now, the concept of FO has mainly been deterministic. More precisely, if we want to model flexibility for a certain device and we are able to determine precisely its status at all times, FOs will either be able to model flexibility accurately, or to give an accurate approximation which is much faster to optimize compared to an exact model. However, in cases where there is uncertainty on the status of the device, only one work has tried to model uncertainty (Frazzetto, et al.), and only related to specific aspects (time of activation for devices).

For this reason, an extension of the FO concept is proposed in this task, which models uncertainty and takes it into account when describing flexibility. At the moment, three types of uncertainty are being considered: *existence uncertainty*, *time uncertainty* and *amount uncertainty*. Existence uncertainty is the uncertainty about the prosumer being actually able to deliver at all the flexibility described in the FO; time uncertainty is the uncertainty about the time at which the prosumer will have the possibility to activate the load, and amount uncertainty is the uncertainty about how much energy can be consumed by the device at a given time. The last two types of uncertainty can be grouped under a more generic concept, called *value uncertainty*.

We want to create a FlexOffer which considers the value uncertainty related to the device. In order to do so, we first need to model this uncertainty. For better clarity, we will consider as a title of example the battery described in the running example, and more specifically the *switching* case. The reason why uncertainty for the battery needs to be modelled is the following: suppose we want to use the battery for flexibility for the following six hours. At the first hour, we know the state of charge of the battery, and therefore the available flexibility is known for certain. At the second hour, however, the amount of available flexibility depends on how much energy has been given to/taken from the battery during the first hour, so there is uncertainty on how much flexibility is available at that time. For the same reason, at the third hour, the amount of available flexibility depends on how much energy has been given to/taken from the battery during the first two hours, and so on.

We know that the amount of flexibility available at each time unit depends on the state of charge *SoC* of the battery. Also, for the sake of simplicity, we will consider this equation for the variation of *SoC(t)*:

$$SoC(t) = SoC(t-1) + u_B(t)$$

$$SoC_{min} \leq SoC(t) \leq SoC_{max} \qquad E_{min} \leq u_B(t) \leq L \cdot E_{max}$$

This is the same as the state equation described before, except $u_B(t)$ is the amount of energy that the battery gives/receives, while $u(t)$ was the amount given to/received from the prosumer. Because of the losses from the charging/discharging processes, those two quantities are different and regulated by the equations:

$$u(t) = \frac{u_B(t)}{L} \text{ if } u(t) \geq 0$$

$$u(t) = L \cdot u_B(t) \text{ if } u(t) < 0$$

3.2. Uncertainty Modelling

Call *T* the time horizon for which we want to model flexibility. Since the amount of available flexibility at time *t* depends on *SoC(t)*, we will have to model uncertainty for *SoC(t)* at each $t \in \{1, \dots, T\}$. *SoC(1)* can be measured, which means that for *t* = 1 we can determine the available flexibility with certainty. However, for *t* > 1, *SoC(t)* depends on the amount of energy used before: therefore, the uncertainty for *SoC(t+1)* depends on the uncertainty for $u_B(t)$.

Suppose that we know the probability distribution function (PDF) that describes the probability for $u_B(t)$ to assume each possible value in $[E_{min}, E_{max}]$, and that we know it for each $t \in \{1, \dots, T\}$: call $\bar{u}_t$ the PDF associated with $u_B(t)$. From this, it is possible to determine the PDF $\bar{u}_{t+1}$ that describes the probability for *SoC(t+1)* to assume each of its possible values in $[SoC_{min}, SoC_{max}]$. Since we already know *SoC(1)*, we have $\overline{SoC}_1 = \delta_{SoC(1)}$, the Dirac delta distribution concentrated at *SoC(1)*. From now on, we can define $\overline{SoC}_{t+1}$ as follows:

$$\overline{SoC}_{\{t+1\}}(x) = \int_{E_{min}}^{E_{max}} \overline{SoC}_t\,(x-r) \cdot \bar{u}_t(r)\,dr$$

This definition is mathematically equivalent to the following: the random variable whose distribution is $\overline{SoC}_{t+1}$ is the sum of the random variable whose distribution is $\overline{SoC}_t$ and the random variable whose distribution is $\bar{u}_t$.

We now know the probability distributions related to *SoC*, and we want to determine the uncertainty in value flexibility. At time *t*, the term $u_B(t)$ indicates the amount by which we will charge the battery: call I_t the set of all values that $u_B(t)$ can possibly assume. We know that I_t is an interval and that $I_t \subseteq [E_{min}, E_{max}]$. Our purpose is to build a function $f_t : [E_{min}, E_{max}] \to [0,1]$ that, for every $x \in [E_{min}, E_{max}]$, describes the probability that $x \in I_t$. Note that f_t is not a PDF: $f_t(x)$ does not describe the probability that $u_B(t)$ will assume the value *x* when the schedule is determined. $f_t(x)$ instead describes the probability that *x* belongs to the set of the feasible values for $u_B(t)$ before the schedule is chosen.

We define f_t as:

$$f_t(x) = \int_{SoC_{min}-x}^{SoC_{max}-x} \overline{SoC}_t\,(r)dr$$

The reason for this definition is the fact that $x \in I$ if and only if $SoC_{min} \leq SoC(t) + x \leq SoC_{max}$, and therefore the probability that $x \in I$ is the probability that $SoC_{min} - x \leq SoC(t) \leq SoC_{max} - x$, which is measured by that integral. Note that f_t is not a probability distribution.

Since it is difficult from a computational perspective to create pointwise-defined functions, we created a discrete approximation approach. We work under the hypothesis that *SoC(t)* and $u_B(t)$ can only assume values which are multiples of *gr* (a number which describes the granularity of the approximation), and therefore $\bar{u}$ and $\overline{SoC}$ are discrete distributions. So, $\overline{SoC}_1$ becomes the zero vector, except for the value $\overline{SoC}_1\big(SoC(1)\big) = \frac{1}{gr}$. The equation for determining $\overline{SoC}_{t+1}$ can be written as

$$\overline{SoC}_{t+1}(x) = \sum_{r=\frac{E_{min}}{gr}}^{\frac{E_{max}}{gr}} \overline{SoC}_t\,(x - gr\cdot r)\cdot \bar{u}_t(gr\cdot r)$$

and consequently, the equation for f_t becomes

$$f_t(x) \;=\; \sum_{r=\frac{SoC_{min}-x}{gr}}^{\frac{SoC_{max}-x}{gr}} \overline{SoC}_t\,(gr\cdot r)$$

An example has been made for the *switching* case of the running example. Here, $\bar{u}_t(x)$ is a uniform distribution with value 0.1 in $x \in [-5{,}0)$ and value 0.105 in $x \in [0{,}4.76]$, and *SoC(1)* = 7 kWh. We chose a value of 0.01 for *gr*.

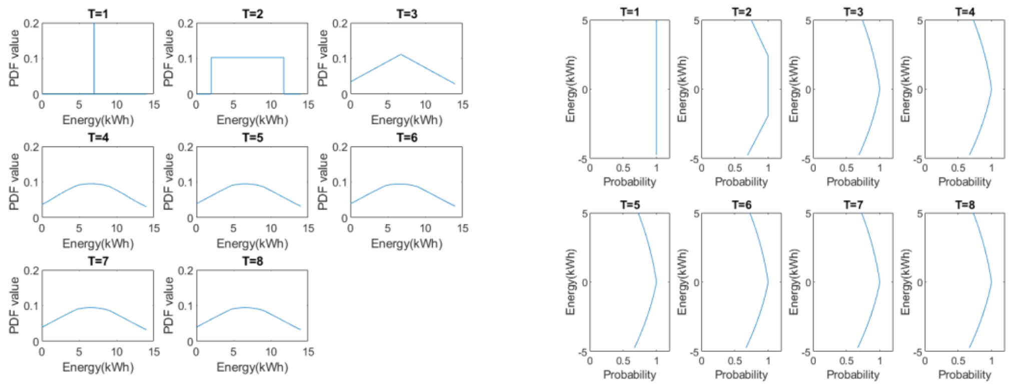

FIGURE 3: PDF OF SOC AND FLEXIBILITY

Figure 3 shows the results for $\overline{SoC}$ and for f_t. The PDFs for *SoC* take different forms. For *t* = 1 we know *SoC(1)* with certainty, so the resulting PDF is a distribution which is zero everywhere except for the value of *SoC(1)*, i.e. a Dirac delta distribution (even if the graph only shows up to 0.2 as value for energy = 7 kWh, it is to be considered as infinity). For *t* = 2 the PDF is a step function. For *t* = 3 the PDF is a continuous function since the values in the defining equation vary continuously, composed by two straight lines: for the same reason, the PDF for *t* = 4 and above is a smooth function, being quadratic for *t* = 4, cubic for *t* = 5 and so on. Those PDFs converge to a limit PDF for $t \to \infty$. About the flexibility probability functions f_t we can see a similar behaviour, with f_t converging to a certain function *f* for $t \to \infty$. It can also be noticed that the values with probability 1 of being available for flexibility are the same values issued for a standard FlexOffer that uses up all the available flexibility right away.

Consider a device we want to model flexibility for, and denote by e_t an amount of energy that this device may consume at time *t*. With the notation used until now, we define an **uncertain FlexOffer** (UFO) *F* as a tuple $\{g_1, g_2 \dots, g_T\}$: here, $g_1 \dots, g_T$ are functions from $\boldsymbol{R}$ to [0,1] such that $g_t(e_t)$ represents the probability for the device to be able to consume the amount of energy e_t at time *t*. With the notation used before, we have

$$g_t(e_t) = P_{et}(F) \cdot f_k(e_k)$$

Where $P_{et}(F)$ is a number representing existence and time uncertainty for *F*.

In order to be optimized, we define a SFO from an UFO, in the following way. Let *pt* be a probability threshold: we want to define a SFO $[emin_1, emax_1, \dots, emin_T, emax_T]$ such that, for every $t \in \{1, \dots, T\}$ and $x \in [emin_t, emax_t]$, we have $g_t(x) \geq \sqrt[T]{pt}$. This way, when multiplying, we ensure that every schedule defined within those constraints has probability *pt* or higher to be feasible.

4. FlexOffers for Other Measures, and Prediction Modelling

4.1. FlexOffers for Other Measures

Up to this point, FOs have only been used for describing energy flexibility. However, it is possible to use them in order to describe other quantities: this is possible because FOs model flexibility for time series, and therefore they can be adapted to model flexibility (and also uncertainty and predictions, as we will see) for other measures, as long as they can be expressed through time series.

In order to explain better this concept, we introduce an example: consider the case of a heat pump. FOs can be used to express the energy flexibility relative to this device; this is achieved by creating a model for the heat pump (i.e., defining temperature thresholds, minimum and maximum energy output), choosing the constraints and type of approximation that we want to use, and then generating the FO. The result will be a set of constraints, describing the constraints for the amount of energy that can be used by the heat pump.

However, using a certain amount of energy will change the temperature of the room. Since temperature can be expressed in a time series, and we know how it changes depending on the amount of energy used, we can generate an FO able to describe the flexibility relative to the room temperature.

There are many more examples that can be represented. For example, a thermal storage unit works with the same principle of a battery, and therefore for this device it is possible to use FOs for representing energy flexibility, but also temperature.

In order to show how FOs can model measures different from energy, we present a very simple example.

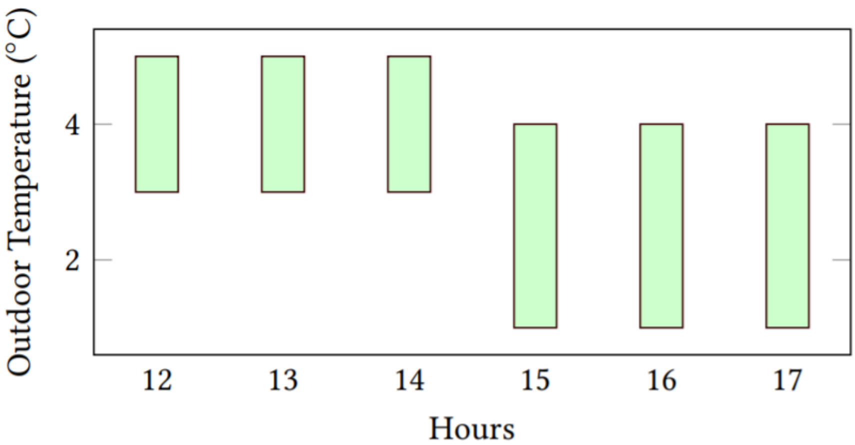

FIGURE 4: FO FOR DESCRIBING OUTDOOR TEMPERATURE

Suppose we want to represent outdoor temperature in Sion, for a time horizon of six hours (12 to 18) in a February afternoon: we have information that in the day before, temperature varied in the range between 3°C and 5°C at every hour between 12 and 15, and between 1°C and 4°C at every hour between 15 and 18. We also know that the weather for the considered day should be the same, and therefore we expect to have a similar pattern for the temperature. A SFO can then represent temperature for the described time horizon, as described in Figure 4: each slice refers to the hour starting at the time it refers to (e.g. the first slice refers to the one-hour time interval starting at 12 and ending at 13, and so on).

We will see another example in the following subsection, where we will represent uncertainty for energy spot prices in order to model predictions for them.

4.2. FlexOffers for Representing Predictions

Another possible use for FOs is to represent predictions. The idea for this concept is to consider an FO representing the measure we want to model, reduce to zero the value flexibility, and eventually consider the uncertainty related to it.

The simplest example that we can use to show this concept, is the battery described in the Running Example section. Suppose we want to represent a prediction for battery usage for the switching case, for example e_1 = 1 kWh, e_2 = 3 kWh, e_3 = 2 kWh, e_4 = e_5 = 0 kWh, e_6 = 1 kWh. This can be easily represented by a SFO with a time horizon of 6 time units, where $emin_1 = emax_1$ = 1 kWh, $emin_2 = emax_2$ = 3 kWh, and so on. Figure 5 shows how such a FO would look like: as the amount of flexibility is zero, it becomes a representation for a time series. In general, FOs can represent in this way any type of time series, and in particular predictions.

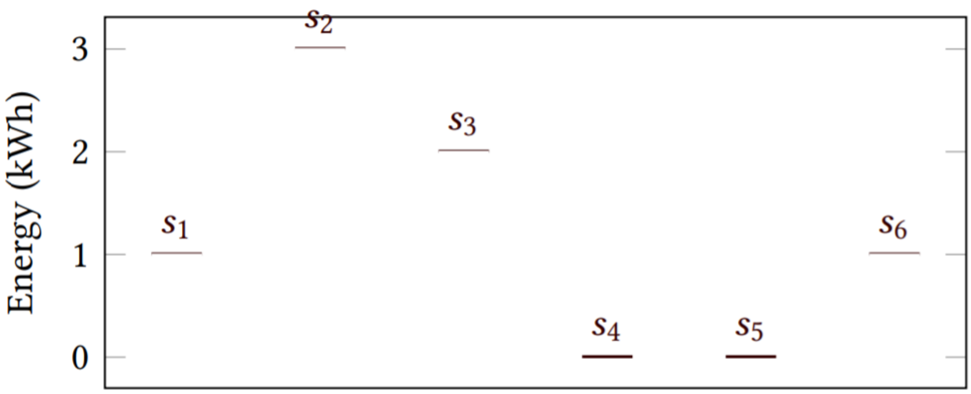

FIGURE 5: A FO MODELLING A PREDICTION FOR BATTERY ENERGY USAGE

A more complex scenario could be a prediction with uncertainty. An example could be as following: suppose we want to predict the price of energy in the spot market for the next hours. Since we have no control over this measure, there is no flexibility related to it. However, we can know some information about the energy price and its uncertainty, and we can use UFOs in order to model that. In this example, we will make two assumptions: first, we want to estimate the price for the first six hours of January 1st for the zone DK1 from NordPool[1], and historical data suggest us a pattern that the price may follow. Second, we are in off-peak hours: this implies that the uncertainty in energy price can be described by a normal distribution (H.Zhou, et al., 2009), whose standard deviation can be considered to be one fifth of

[1] https://www.nordpoolgroup.com/

the expected value. With these premises, it is possible for us to generate an UFO which models the predictions for the price: this can be seen in Figure 6.

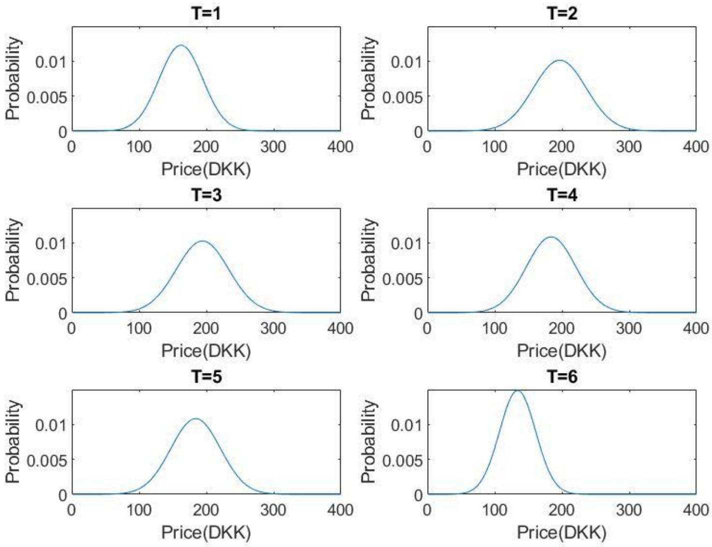

FIGURE 6: AN UNCERTAIN FO FOR PREDICTING SPOT ENERGY PRICPES

5. Demonstration

In this section we will describe our experiments for determining the functioning of UFOs, and the code that has been written to implement them.

5.1. Experimental evaluation

To evaluate the performances of UFOs, we have run some experiments and measured the amount of provided flexibility. There are several metrics that can evaluate flexibility (Valsomatzis, et al., 2015), but in a real case, one of the most important is economic revenue (Lilliu, et al.). This metric is defined by the function $profit(e) = prices \cdot e$, where $\mathrm{e} = (\mathrm{e}_1, \dots, \mathrm{e}_\mathrm{T})$ expresses the energy consumed at each time $1, \dots, T$ and $\mathrm{prices} = (\mathrm{prices}_1, \dots, \mathrm{prices}_\mathrm{T})$ represents the prices for energy at time 1, …, T. Depending on which market we are operating in, *prices* can be either the spot or imbalance prices.

We simulated the battery B from the running example. Data for prices have been taken from a sample in NordPool, and include spot prices and imbalance prices between January 1, 2018, and December 31, 2018, for the zone DK1. Spot prices will be used for the *profit* function, while imbalance prices for calculating imbalance penalties, in case the schedule obtained by optimization is unfeasible. The battery works in the following context: prices are known in advance from the day-ahead market. When the FO is issued and optimized, but before it is executed, a bid for buying/selling energy is made, according to those prices: for simplicity, we assume that the bid is always accepted. The prosumer will then give/receive money according to it, as defined by the *profit* function calculated with the spot prices. However, when the schedule is executed, if the prosumer is unable to fulfil the bid (e.g., tries to sell energy when the battery is empty, or to buy it when the battery is full), this will generate imbalance in the grid, and the prosumer will have to pay for this imbalance.

Our experiment works as follows. We start with *B* with the same settings of the *charging* example, and we choose a time horizon *T* and a probability threshold *pt* for generating and optimizing UFOs. Now, we want to issue two FOs in succession: one for charging the battery for the first *T* time units, and one for discharging it in the following *T* time units. We first estimate the value for *SoC(T)* that would maximize the *profit* function over the next *2T* time units: call this value SoC_{MP}. After that, we generate two UFOs: one for charging the battery, with *SoC(0)* = 0 kWh, and one for discharging it afterwards, with *SoC(0)* = SoC_{MP}, as *SoC(0)* for the second FO is equal to *SoC(T)* for the first FO. We then optimize those FOs, with the objective to maximize the profit function: *prices* is defined by the spot market prices data and *e* being the energy variable, and the constraints on *e* are defined by the FO. We then check whether the schedules obtained by the optimization violate the constraints of the battery model; in case they do, we also calculate the penalty for violation. We calculate it as the minimum possible cost of the difference between the schedule and a feasible one, calculated as the negative *profit* functions, where *prices* are imbalance prices. We then repeat this procedure for the next *2T* time units again and again, until the simulation covers a total of 365 days. This experiment is also performed with two other approaches, which will be the baselines: inner approximation SFOs, and an exact approach based on the LTI model (Lilliu, et al.).

We have run this experiment for *T* = 4 and different values of *pt*. Results are reported in Table 1.

TABLE 1: RESULTS FOR ECONOMIC PROFIT (IN DKK)

	Profit (before imbalance calc)	Imbalance Costs	Profit (after imbalance calc)
LTI	466349	0	466349
Inner SFOs	194547	0	194547
UFOs, pt = 1	195137	111	195026
UFOs, pt = 0.95	354853	62405	292448
UFOs, pt = 0.9	397586	109885	287701
UFOs, pt = 0.85	407356	129752	277603

From the results, some important observations can be done. First, for *pt* = 1, UFOs behave almost exactly like inner SFOs, and the difference comes from numerical error on probability. Second, for lower values of *pt*, UFOs perform better than inner SFOs in terms of profits, even after considering imbalance penalties. In particular, inner SFOs are able to retain 41.8% of the total profits, while UFOs vary depending on *pt*, and for *pt* = 0.95 are able to retain 62.7% of the total profits. It has to be noted that for lower values of *pt*, this amount decreases: this is because imbalance penalties become too high, and make the profits decrease despite the higher amount of flexibility.

An experiment has also been run in order to compare optimization speed between those three approaches (Lilliu, et al.). Results show that optimization time grows exponentially for the exact model, and for *T* = 24 optimization is performed 4.7 hours, making it unfeasible in practice; conversely, inner SFOs take at most 0.103 seconds for that amount of time, and the same is true for UFOs, as the type of constraint for optimization is the exact same as SFOs, and therefore the difference in optimization time is minimal, lower than 0.01 seconds in all cases. We have also measured the amount of time needed for generating the slices: for *T* = 24, it takes an amount of time lower than 0.055 seconds. Figure 7 shows this more in detail.

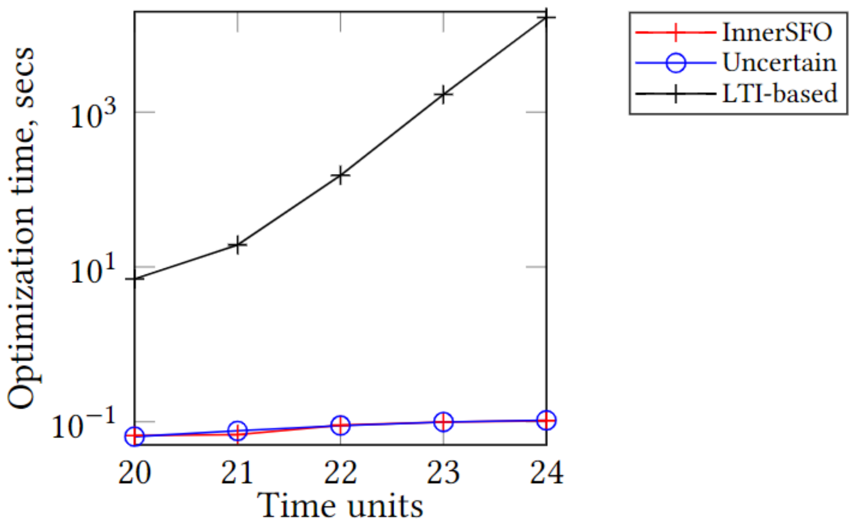

FIGURE 7: OPTIMIZATION TIME FOR FOs

5.2. Code

All the code used for implementing the FOs described in this deliverable is available at https://github.com/FabioLilliu/domOS-AAU/ . All the files are in the MATLAB language. The following files are contained:

- BasicBatteryFlexOfferTester: generates FlexOffers for a battery, given the desired battery specifics and type of approximation.
- Const_prob,const_AAO: generates constraints for UFOs and inner standard FOs respectively.
- Const_prob_charging, const_AAO_chargin: generates constraints for UFOs and inner standard FOs respectively, for the *charging* case.
- Experiment: reproduces the experiment of Section 5.1 for maximizing the profit function.
- ExperimentForOptimizationTime: reproduces the experiment for measuring optimization time.
- Generate_discrete_uncertaintyFO: generates an UFO for a battery, given the desired battery specifics.
- Imbalance_calculation: function that calculates violation penalties as described in Section 5.1.
- IndoorTemperatureExample: describes the example shown in Section 4.1 for modelling temperature with FOs.
- PricePrediction: generates an UFO for predicting spot prices, similar to what shown in Section 4.2.
- prob_slices: function that receives as an input battery data, time horizon and probability threshold, and returns an UFO.

In addition to this, the repository will also contain the code for the FlexOffer Agent (FOA) described in D4.5, which also includes the basic FO stack over which the work described in this section has been built. The FOA allows to receive load forecasts represented by time series and generate FOs from three different types of devices (wet devices, thermostatically controlled devices, battery devices) following two main schemes (individual, and pool-based). Further information about it can be found in D4.5.

6. Conclusion

This document presents the work done on task T3.3, specifically related to the extension and generalization of FlexOffers in order to i) be able to capture predictions, flexibility and uncertainty; ii) to be able to do so not only for energy, but also for other measures such as temperature and energy prices. We first defined the concept of FlexOffer, describing its life cycle and showing how it models flexibility. We then extended this concept in order to be able to model uncertainty by introducing uncertain FlexOffers and, after that, showed how they can be used to model predictions, and how they can be employed to model measures different from energy. Finally, we showed their validity through experiments, and provided the code needed to recreate the same examples and experiments. Future work in task 3.3 will be focused on a modelling methodology and guide for modelling demonstration sites assets, and on the integration of FlexOffers into the domOS Common Ontology and will be reported in the following WP3 deliverables.

7. References

Böhm Matthias [et al.] Data management in the MIRABEL smart grid system [Tidsskrift]. - [s.l.] : EDBT/ICDT workshops, 2012.

Borrelli Francesco, Bemporad Alberto og Morari Manfred Predictive Control for Linear and Hybrid Systems [Tidsskrift]. - [s.l.] : Cambridge University Press, 2017.

Corsetti Edoardo [et al.] Modelling and deploying multi-energy flexibility: The energy lattice framework [Tidsskrift]. - [s.l.] : Advances in Applied Energy, 2021.

Ferreira Luis Lino [et al.] Arrowhead compliant virtual market of energy [Tidsskrift]. - [s.l.] : IEEE ETFA, 2014.

Frazzetto Davide [et al.] Adaptive User-Oriented Direct Load-Control of Residential Flexible Devices [Tidsskrift]. - [s.l.] : e-Energy 2018.

H.Zhou [et al.] Study on probability distribution of prices in electricity market: A case study of Zhejiang province, China [Tidsskrift]. - [s.l.] : Communications in Nonlinear Science and Numerical Simulation, 2009.

Junker Rune Grønborg [et al.] Characterizing the energy flexibility of buildings and districts [Tidsskrift]. - [s.l.] : Applied Energy, 2018.

Junker Rune Grønborg [et al.] Stochastic nonlinear modelling and application of price-based energy flexibility [Tidsskrift]. - [s.l.] : Applied Energy, 2020.

Koller Michael [et al.] Defining a degradation cost function for optimal control of a battery energy storage system [Tidsskrift]. - [s.l.] : 2013 IEEE Grenoble Conference.

Lilliu Fabio, Pedersen Torben Bach og Siksnys Laurynas Capturing Battery Flexibility in a General and Scalable Way Using the FlexOffer Model [Tidsskrift]. - [s.l.] : SmartGridComm 2021.

Meng Jinhao [et al.] A Simplified Model-Based State-of-Charge Estimation Approach for Lithium-Ion [Tidsskrift]. - [s.l.] : TIE, 2019.

Neupane Bijay, Pedersen Torben Bach og Thiesson Bo Utilizing Device-level Demand Forecasting for Flexibility Markets [Tidsskrift]. - [s.l.] : e-Energy 2018.

Nguyen Jennifer L, Schwartz Joel og Dockery Douglas W. The relationship between indoor and outdoor temperature, apparent temperature, relative humidity, and absolute humidity. [Tidsskrift]. - [s.l.] : Indoor Air, 2014.

Pedersen Torben Bach, Siksnys Laurynas og Neupane Bijay Modeling and Managing Energy Flexibility Using FlexOffers [Tidsskrift]. - [s.l.] : SmartGridComm, 2018.

Schott Paul [et al.] A Generic Data Model for Describing Flexibility in Power Markets [Tidsskrift]. - [s.l.] : Energies, 2019.

Siksnys Laurynas og Pedersen Torben Bach Dependency-based FlexOffers: scalable management of flexible loads [Tidsskrift]. - [s.l.] : e-Energy, 2016.

Valsomatzis Emmanouil [et al.] Measuring and Comparing Energy Flexibilities [Tidsskrift]. - [s.l.] : EDBT/ICDT, 2015.