This document constitutes a reference description form to facilitate and set a standard for information exchange between stakeholders on the definition of a **control function (CF)**. The CF is a description of an embedded control system (or an aspect thereof) which, together with the inherent physical properties of the system itself, defines the behavior of the control system and its response to dynamic input. Control functions can be seen as a further detailing, or more formal description, of the mechanisms governing the behavior of individual actors in a use case.

Control functions may take the form of mathematical functions, such as transfer functions or descriptions of a stochastic process, or they may be best described by an algorithm. These different forms require different description options. As a result, some of the sections of the document are optional because their content does not apply to all descriptions. This is indicated at the beginning of each section. Optional sections should be left blank, rather than being deleted entirely, in order to preserve the consistency of section numbers. Sections marked as mandatory must be filled in every case.

NOTE: This document has been carefully structured in a way that allows to extract the contained information in an automated way and to translate it to other machine-readable formats (parsing and conversion by script). Therefore, when editing this document, please keep the following in mind:

* Only information (text and figures) provided in the cells of the tables defined below will be extracted.
* The first cell of each table contains a specific title that identifies its purpose ("Document History", "Functional Description", etc.). Do NOT change these titles, even if you replicate a table (as in the case for the control function inputs).
* When extending tables (e.g., add more instances to an element description), insert new row with the same layout. Especially do NOT split or merge cells in a table’s row.
* You may add additional (sub-)headings to structure the document to accommodate your needs.
* Names and identifiers should be machine-readable, i.e., no spaces, no special symbols, only basic ASCII letters and numbers.

1. Document History

Please keep track of significant changes to this document. Consider using the following versioning scheme to indicate the stage of the CF: 0.x – concept stage, 1.x – test design stage, 2.x – post test / documentation stage, 3.x and higher – revisiting the test.

|  |  |  |  |
| --- | --- | --- | --- |
| Document History | | | |
| Date | Version | Description | Author |
| 2021-05-13 | 0.1 | first draft | Evangelos Rikos (CRES) |
| 2021-05-18 | 0.2 | update of diagrams and control concept based on discussion with partners | Evangelos Rikos (CRES) |
| 2021-11-02 | 1.0 | update controller specification to represent the final implemented benchmark setup | Edmund Widl (AIT) |
| 2024-25-01 | 1.1 | Added identification | Filip Pröstl Andrén (AIT) |

1. Control Function Identification

Please specify a name for this CF together with an ID to identify it.

NOTE: Please use a machine-readable ID.

|  |  |
| --- | --- |
| Control Function Identification | |
| Control Function ID | Name |
| MENB-CF-VC | Voltage controller for the multi-energy network benchmark |

1. Functional Description

The purpose here is to provide a brief introduction to the problem solved by the function or the algorithm. The technical background should be outlined by giving an overview of the function or algorithm. It will be important in the early introduction of the document to clearly inform the reader on the intent of the function or algorithm as well as the problem the function or algorithm is addressing. The author should keep in mind that this document is not a technical paper and should be written in such a way that it can be understood by an audience with limited expertise in specific simulation domains.

The statement of the problem should include a clear statement outlining the problem and why the problem is important and needs to be addressed. It should contain a sketch of the problem/situation and, if known and/or relevant, the background history that led to the development of the function or algorithm, including why the problem was investigated and how it relates to earlier work that has been done in the field. It should show that the author thoroughly understands the problem.

|  |
| --- |
| Functional Description |
| Background:  System configuration MENB-SC specifies an electrical network, including a voltage controller, see Figure 1. The system comprises two PV systems, two (aggregated) electricity consumers, a heat pump and a simplified electrical distribution network consisting of two lines and two buses. The distribution grid is connected to the external electrical grid (equivalent voltage source).  This specific setup is typical for a local energy community (specifically a renewable local energy community) and, as such, allows to address the issue of exploiting the local PV generation in a way that avoids injection of extra energy to the upstream network (self-consumption). In order for an energy community to be autonomous, the size of the PV systems is such that at times high power generation may occur. However, high power generation from the PVs with no corresponding high consumption on the consumer side can lead to a significant voltage rise in parts of the grid (e.g., busbar bus\_1). Therefore, synchronization of consumption with generation is necessary to ensure the power quality and avoid any disruption due to overvoltage limit violations.    Figure 1: Electrical sub-system including the voltage controller  Problem formulation:  The voltage controller specified in this document is intended to prevent overvoltages at busbar bus\_1 due to high PV generation and simultaneous low power consumption by governing the operation of the heat pump accordingly. In the same way, it should prevent undervoltages due to increased power consumption and reduced PV generation.  Controller scope:  In this context, the scope of this controller is to monitor the voltage at busbar bus\_1 and increase or decrease the power consumption of the heat pump (controllable/flexible load) in order to keep the voltage within an allowed range. |

1. Terminology (optional)

The role of terminology is important, and it defines concepts used in the scientific document. The definitions should be made precise, concise, and unambiguous. Each term should be used in one and only one way throughout the document.

|  |  |
| --- | --- |
| Terminology | |
| Local energy community | Local energy communities (LEC) are a way to organize collective energy actions around open, democratic participation and governance and the provision of benefits for the members or the local community (Roberts et al., 2019). |
| Renewable local energy community | A special type of LEC that incorporate renewable energy sources, renewable local energy communities can be active in all energy sectors and involve activities in production, consumption and selling of renewable energy. |
| Flexible load | A load that has the capability of adjusting its consumption based on specific optimality criteria and/or control signals. |

1. Methodology

This section is used to provide the methodology or theory of the function or algorithm in detail. Relevant research results should be referenced. The description should not be as detailed as a peer-review journal paper. However, all information necessary for understanding specifics of the function or algorithm, especially for translating it into software, must be included. Required equations and computations should be described and explained term by term and be closely related to the detailed description of the function or algorithm later in the document.

In case of well-known or standard functions or algorithms, a simple reference to the theory (e.g., a link to the Wikipedia page for "PID controller") is sufficient.

|  |
| --- |
| Methodology |
| The rationale of the applied control methodology is based on the droop control approach used for distributed energy resources (DER) to support voltage variations (see also Figure 2):   * The voltage at busbar bus\_1 is periodically monitored (with a time resolution depending on the simulation setup, see test specification MENB-TS01). * The measured value of the bus voltage is compared to (subtracted from) the reference value (nominal voltage) to determine the deviation (error). * In case the measured voltage is within the allowed voltage range, no action is taken by the controller. * Otherwise, in case the voltage error exceeds the allowed voltage range, a new setpoint for the heat pump’s power consumption is calculated to compensate for the deviation (see Figure 2 and definition of applied algorithm below). |

1. Limitations

This section describes prerequisites for the function or algorithm. These include limitations caused by input data, special types of data required, limitations inherent to the function or algorithm, situations where the function or algorithm cannot be applied, potential implementation difficulties, and any other problems known to be associated with the function or algorithm. This section should also describe both the implementation and operational impacts of these limitations.

|  |
| --- |
| Limitations |
| Heat pump consumption  A major limitation in the operation of this control scheme is the maximum/minimum power that the heat pump can consume. In several cases of implementing the droop control technique it is possible that the required setpoint violates the operating limits of the controlled component. Therefore, in the selected implementation two limitations are considered:   * The controller cannot request a power consumption from the heat pump above a certain limit (electrical power rating of compressor), regardless of the network condition. * The controller cannot request a power consumption from the pump below a certain limit (minimal electrical power consumption for operating the compressor). However, the controller can request the heat pump to be switched off (set setpoint to 0).   Operation veto  The control signals must not result in an actuation pattern that switches the heat pump on and off in rapid succession. Therefore, after switching the heat pump on, the voltage controller must not switch off the heat pump for a certain amount of time, regardless of the network conditions. Vice versa for switching the heat pump off. This is referred to as operation veto.  Time resolution  Another important limitation is the time resolution selected for the simulation and, consequently, the control loop. In an actual system, voltage variations may take place at very small time scales (<1s). In this case, however, slow (quasi-static) variations will be considered (i.e., in the order of minutes). Essentially, the specified voltage controller can be regarded as a tertiary voltage control scheme dealing with the slow (long-term) variations rather than the short-term dynamics.  Output signal granularity  The granularity of the output signal may be a limitation in view of the precision of the voltage control. The use of a discretization function leads to an approximation of the optimal value that would be required at every time instance. To ensure sufficient granularity, the right amount of discretization steps must be selected in advance, depending on the maximum power range of the heat pump. However, since this control scheme aims to contain voltage deviations and not to precisely compensate them, fine granularity is not crucial.  Response dynamics  A limitation factor can potentially be the time response of the heat pump after the setpoint changes. However, since a quasi-dynamic approach is assumed, the fast dynamics of the network voltage are neglected. |

1. Inputs

Please replicate the table below for each control input. This table describes data inputs to the documented function or algorithm. This includes static input data such as configuration or parametrization as well as dynamic input data such as measurement data or events. Each input data element should specify the data type (e.g., floating point, integer, character string), units (e.g., degrees, percentages), valid ranges (e.g., 0 to infinity; -1.0 to +1.0 inclusive), the expected update rate of the data (e.g., static; 10Hz; daily) and a short description of the information the variable contains (e.g., “width of the radar beam in degrees …”).

All data element names should be self-descriptive (i.e., the name should describe what the data element contains). Widely recognized symbols from textbooks and professional literature are acceptable.

NOTE: Please use machine-readable names.

|  |  |
| --- | --- |
| Control Function Input | |
| Name | meas\_voltage\_pu |
| Type (according to SC description) | MeasureConnection |
| Unit | p.u. |
| Range | [0.8, 1.2] |
| Expected update rate | 1-15 min (simulation step size) |
| Description | voltage at busbar bus\_1 as measured by the voltage sensor (RMS voltage transducer) |

1. Outputs

Please replicate the table below for each control output. This table describes data output produced by the documented function or algorithm, insofar as the output is relevant to the interaction between the function or algorithm and the rest if the system. This covers mostly dynamic output data such as setpoints or events. Each output data element should describe the format of the output data in sufficient detail for downstream algorithms to be easily interfaced. This should as a minimum include the data type (e.g., floating point, integer, character string), units (e.g., degrees, percentages), valid ranges (e.g., 0 to infinity; -1.0 to +1.0 inclusive), the expected update rate of the data (e.g., static; 10Hz; daily) and a short description of the information the variable contains (e.g., “power setpoint in kW”).

All data element names should be self-descriptive (i.e., the name should describe what the data element contains). Widely recognized symbols from textbooks and professional literature are acceptable.

NOTE: Please use machine-readable names.

|  |  |
| --- | --- |
| Control Function Output | |
| Name | setpoint\_hp\_p\_el |
| Type (according to SC description) | CtrlConnection |
| Unit | MWel |
| Range | [0, 0.1] |
| Expected update rate | 1-15 min (simulation step size) |
| Description | proposed heat pump setpoint for electrical consumption, sent to flex heat controller |

1. Use Cases

Use Cases are a way to express functional requirements. They bridge the gap between user needs and system functionality by directly stating the user’s intention and system response for each step in a particular interaction. No single Use Case specifies the entire requirements of the system. A Use Case itself is an interaction that a User or another System has with the system that is being designed to achieve a goal. Different scenarios within a Use Case show how the goal succeeds or fails. A success scenario is one in which the goal is achieved; a failure scenario is one where the goal is not achieved. Because the goals summarize the intention of the various uses of the system, the users can see how they are supposed to use the system. Users can also spot when the system does not support all of their goals without having to wait for the first prototype or having to wait for the system to be developed.

This is a summary of how to write Use Cases:

* The name of the use case should start with a strong verb.
* A Use Case is a set of scenarios. A scenario is a list of steps.
* Each step should state what the user does and/or how the system responds.
* The steps must not mention how the system does something.
* Each step needs to be analyzed in detail before it becomes code.
* Keep It Simple: use the simplest format you need.
* Keep track of different versions.
* Writing use cases is a team sport.
* Focus on a particular user (give them a name).

Don't get bogged down in all the special ways it can go wrong until you've finished the main success story.

|  |  |
| --- | --- |
| Use Case Example | Voltage support at distribution grid substation |
| Date created | 2021-05-13 |
| Actor | * voltage sensor (RMS voltage transducer) at busbar bus\_1 * flex heat controller (controls the heat pump operation) * heat pump |
| Description | The voltage controller modifies the setpoint of the heat pump’s active power consumption to contain/reduce the voltage deviation at busbar bus\_1. |
| Preconditions | Setpoint for heat pump power consumption is within operational limits and there is no operational veto pending. |
| Postconditions | The bus voltage is kept within the limits of operation. |
| Priority | medium |
| Frequency of use | periodically (according to simulation step size) |
| Normal course | The controller measures the voltage deviation. If the deviation is within the allowed voltage range, no action is taken. |
| Alternative course | The controller measures the voltage deviation. If the deviation is not within the allowed voltage range, a new setpoint is calculated and sent to the flex heat controller. The flex heat controller takes this setpoint into consideration when actuating the hydraulic pump in the heat pump’s condenser loop. |
| Exceptions | N/A |
| Assumptions | N/A |
| Notes and issues | N/A |

1. Diagrams (data flow diagrams, sequence diagrams, logic diagrams, state diagrams, control hierarchy, etc.)

This section will contain diagrams which capture the data flow occurring between a function or algorithm and the rest of the system. These diagrams should reveal relationships among and between the various components in a program or system.

In case of distributed controllers, i.e., multiple processes communicating to achieve an overall objective, sequence diagrams of the interaction between the individual processes must be provided here, in addition to the detailed descriptions of the individual processes in subsequent sections.

Where applicable, this section should also contain Logic Diagrams and/or State Diagrams. Logic Diagrams represent logical concepts and State Diagrams represent the behavior of a system. State diagrams describe all the possible states of an object as events occur. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system.

NOTE: Please put each drawing in a separate table cell. Each table cell with a drawing should be preceded by a table cell containing the drawing’s title.

|  |
| --- |
| Diagrams |
| Schematic view of the voltage control algorithm |
| Figure 2: Schematic view of the voltage control algorithm |

1. Algorithmic Functions (pseudocode)

**This section is optional; however, at least one of the three detailed description sections (Deterministic Functions, Stochastic Functions, Algorithmic Functions) must be filled in.**

This section will contain the description of algorithms in a step-by-step fashion, preferably using pseudocode if applicable.

The term “pseudocode” is used here to describe an English-like language that articulates the steps carried out in an algorithm. It allows the author to focus on the logic of the algorithm without being distracted by details of any given programming language’s syntax. For the sake of completeness and consistency, pseudocode should follow a general syntax convention (see the appendix). Pseudocode can be augmented with natural language where convenient.

In case of distributed controllers, i.e., multiple processes communicating to achieve an overall objective, sequence diagrams of the interaction between the individual processes must be provided in the "Diagrams" section, in addition to the detailed descriptions of each of the individual processes here.

NOTE: Please put each algorithm in a separate table cell. Each table cell with an algorithm should be preceded by a table cell containing the algorithm’s title.

|  |
| --- |
| Algorithms |
| Voltage controller algorithm |
| **BEGIN ALGORITHM** (voltage controller)  **DETERMINE** operation veto still pending  **IF** operation veto pending **THEN**  **EXIT** **ALGORITHM** (voltage controller)  **END** **IF**  **READ** *meas\_voltage\_pu* // get latest voltage measurement  **READ** *setpoint\_hp\_p\_el* // get latest value for setpoint  // lower voltage band threshold depends on heat pump status  **IF** heat pump is switched off **THEN**  **SET** *delta\_vm\_lower\_pu* = *delta\_vm\_lower\_pu\_hp\_off*  **ELSE**  **SET** *delta\_vm\_lower\_pu* = *delta\_vm\_lower\_pu\_hp\_on*  **END** **IF**  **COMPUTE** *delta\_v\_meas\_pu* = *meas\_voltage\_pu* - 1  **IF** *delta\_vm\_lower\_pu* < *delta\_v\_meas\_pu* < *delta\_vm\_upper\_pu* **THEN**  **IF** heat pump is switched off **AND NOT** operation veto pending **THEN**  **SET** *setpoint\_hp\_p\_el* = *hp\_p\_el\_mw\_min*  **SET** operation veto to pending  **END** **IF**  **ELSE**  // update setpoint  **COMPUTE** *res* = *k\_p* \* (*delta\_v\_meas\_pu* - *delta\_vm\_deadband*) / *hp\_p\_el\_mw\_step*  **COMPUTE** *step\_res* = int(*res*)  **IF** fabs(*res* - *step\_res*) > *hp\_p\_el\_mw\_step* **THEN**  **SET** *setpoint\_hp\_p\_el* = *setpoint\_hp\_p\_el* + *hp\_p\_el\_mw\_step* \* (*step\_res* + 1)  **END** **IF**  // check operational constraints  **IF** new setpoint above max. heat pump consumption **THEN**  **SET** *setpoint\_hp\_p\_el* = *hp\_p\_el\_mw\_rated*  **ELSE IF** new setpoint below min. heat pump consumption **AND** \  **NOT** nooperation veto pending **THEN**  **SET** *setpoint\_hp\_p\_el* = 0  **SET** operation veto to pending  **ELSE IF** new setpoint below min. heat pump consumption **AND** operation veto pending **THEN**  **SET** *setpoint\_hp\_p\_el* = *hp\_p\_el\_mw\_min*  **END IF**  **END IF**  **END** **ALGORITHM** (voltage controller) |

1. Deterministic Functions

**This section is optional; however, at least one of the three detailed description sections (Deterministic Functions, Stochastic Functions, Algorithmic Functions) must be filled in.**

This section will contain a mathematical description of the deterministic (as opposed to stochastic) part of a function. Each mathematical formula should be followed by a brief explanation of the symbology.

The section can also be used to document the mathematical background of an algorithm for which pseudocode is provided in the "Algorithms" section below. In this case, the description should also cover the method used to realize the implementation in the Algorithms section.

|  |
| --- |
| Deterministic Functions |
| N/A |

1. Stochastic Functions

**This section is optional; however, at least one of the three detailed description sections (Deterministic Functions, Stochastic Functions, Algorithmic Functions) must be filled in.**

This section will contain a mathematical description of the stochastic (as opposed to deterministic) part of a function. Each mathematical formula should be followed by a brief explanation of the symbology.

The section can also be used to document the mathematical background of an algorithm for which pseudocode is provided in the "Algorithms" section below. In this case, the description should also cover the method used to realize the implementation in the Algorithms section.

|  |
| --- |
| Stochastic Functions |
| N/A |

1. Deployment (optional)

**This section is optional. It should be filled in if the described function or algorithm is deployed in a concrete setup (e.g., as part of a lab experiment).**

This section provides additional information regarding the deployment of the control function in a hardware and/or software setup.

|  |
| --- |
| Deployment (optional) |
| N/A |

1. References

|  |
| --- |
| References |
| [1] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodriguez, “Control of Power Converters in AC Microgrids”, IEEE Trans. Power Electron., vol. 27, no. 11, pp. 4734-4749. Nov. 2012. |