Ref. Ares(2015)6002506 - 22/12/2015

Unifying Control and Verification of Cyber-Physical Systems (UnCoVerCPS)

WP5 Realisation of Cyber-Physical Systems (Tasks 5.1,5.2,5.3,5.4)

D5.1 – Report on Application Models

WP5	**D5.1 – Report on Application Models**
Authors	Javier Sanchez Cubillo (Tecnalia R&I), Simone Schuler (GE Global Research), Daniel Hess (DLR), Maria Prandini (PoliMi) and Mark Burgin (RUR)
Short Description	Description of application models of the use cases (wind turbine, smart grid, automated vehicle, human-robot co-working) used for demonstrating the novel on-the-fly verification methods of cyber-physical systems in UnCoVerCPS.
Keywords	Mathematical modeling, wind turbine, smart grid, automated vehicle, human-robot co-working.
Deliverable Type	Report
Dissemination level	Public
Delivery Date	31 Dec 2015
Contributions by	Arkaitz Urquiza, Gorka Lorenzo (Tecnalia R&I)
Internal review by	Xavier Fornari (Esterel Technologies) and Mark Burgin (RUR)
External review by	
Internally accepted by	Matthias Althoff
Date of acceptance	22 Dec 2015

Document history:

Version	**Date**	**Author/Reviewer**	**Description**
1.0	14/09/2015	Javier Sanchez Cubillo	First draft version
1.1	30/09/2015	Javier Sanchez Cubillo	Inclusion of models
1.2	06/10/2015	Javier Sanchez Cubillo	Reviewable version
1.3	09/11/2015	Javier Sanchez Cubillo	Reviewed version
1.4	10/12/2015	Javier Sanchez Cubillo	Final version

Contents

1 Introduction

Within UnCoVerCPS we develop novel methods to reason about safety of cyber-physical systems in known and unknown environments. We consider both, design (i.e. offline) and operational (i.e. online) verification and validation of these systems. Verification is understood here in a formal way, i.e. we want to reason beyond doubt that a system satisfies desired properties under all conditions. To demonstrate that the novel methods developed within UnCoVerCPS can be applied to a broad range of cyber-physical systems, we selected four illustrative use cases: wind turbines, automated driving, smart grids and physical human-robot interaction. As can be noted, we carefully selected use cases from completely different industrial sectors, however, they all operate in safety critical environments. By applying the solutions developed within UnCoVerCPS, we show that the behaviour of these apparently different applications can be verified by a common and unified approach.

The *wind turbine* is a nonlinear system operating in unknown environmental conditions (mainly the wind acting on the rotor). It exhibits a switching dynamic behaviour based on the current wind conditions. The main goal in wind turbine control is the increase of power production and the reduction of mechanical stress on the structure. Wind turbines also have to enforce stringent safety conditions. These requirements are not related to the physical or computational part only, but rather a property of the whole system.

Smart grids are large-scale power systems consisting of varying numbers of generating, transmitting and consuming nodes. In additions they are exposed to a variety of stochastic uncertainties. They might exhibit a switching topology, due to renewable energy generators entering or leaving the gird. Another crucial aspect is the robust, fault-tolerant, and stable smart grid operation in case of faulty operation of individual grid components.

In the case of *verified automated driving*, safety guarantees for automated driving and advanced driving assistance systems are important for the acceptance in the general public. Automated driving has to cope with diverse conditions and requirements, coming from inaccurate measurements, internal delay, behavioural prediction of other traffic participants and the cooperation and communication with them. Within UnCoVerCPS we focus on collaborative automatic lane change manoeuvres and interaction between human drivers and advanced driver assistance systems.

In *human-robot collaboration*, a tight physical interaction between humans and intelligent machines takes place. For a human worker to collaborate with a robot co-worker in overlapping work spaces, safety guarantees have to be ensured to protect the human and any contact between the robot and the human has to be avoided. A key element is the ability of the robot to adapt its behaviour (slow down, choose another target and / or abandon the current target) to the foreseen human motion, thus avoiding potentially harmful situations. However, we do not want the robot to act too conservative around the human, such that the collaboration suffers. Within UnCoVerCPS we specifically look at humans and robot collaborating on food assembly for e.g. pizzas or sandwiches.

Each of these application scenarios has unique features, being it the scalability and stochastic nature of the smart grids, the unknown environmental conditions and switching behaviour in case of the wind turbine, the human factor in case of the human-robot collaboration or the safe autonomous behaviour in case of the automated driving. Within all four use cases, the systems are exposed to (partially) unknown conditions that might put them or their environment into hazardous situations or raise variables like consumption or cost unpredictably. However, within UnCoVerCPS, we aim to guarantee that system specifications are met, even in unknown situations, either by offline verification or by online re-design of the controller.

To analyse and verify the behaviour of a system, a mathematical description in form of dynamical and/or static equations is needed. The tools and methods for verification and validation developed within UnCoVerCPS are then applied to these models to predict the behaviour of the real system and to give safety guarantees that hold true even in unknown environmental conditions. Therefore, within this deliverable, we give a detailed introduction to the four use cases, describe the mathematical modelling formalism and, if appropriate, explain state of the art control algorithms. Description on how these models are implemented and first simulation results complete the modelling. These models will enable all project partners to test, validate and compare the novel methods developed during the course of UnCoVerCPS.

The remainder of this report is structured as follows: Detailed mathematical models of the four use cases are described in Section 2 for the wind turbine, Section 3 for the smart grid, Section 4 for the automated driving and Section 5 for the human-robot collaboration and the report concludes in Section 6 with a summary and an outlook. In addition, implementation and preliminary simulation results are shown in Appendx A.

2 Modelling of a wind turbine as a hybrid system

The first application area within UnCoVerCPS is a wind turbine. This chapter describes the physical background of power capture from wind, the nonlinear modelling of the wind turbine and its actuator dynamics. State of the art control algorithms are presented and a combined hybrid system, consisting of turbine and controller is presented. As will be seen, the turbine itself is not a hybrid system, however, the way we control the power production and load mitigation of wind turbines makes a hybrid controller necessary.

2.1 Power capture in wind turbines

The maximum power that can be theoretically extracted from the wind is given by

$$P = \frac{1}{2} c_p \rho A v^3, \tag{1}$$

where c_p is the power coefficient, ρ is the air densitiy, A is the area covered by the rotor and v is the wind speed. Wind power is thus proportional to the area covered by the rotor blades and proportional to the third power of the wind speed (see also Figure 1, dashed line). Thus, there are two possibilities to extract more power from the wind, either by using larger turbine rotors, or by installing the turbines at places with higher average wind speed. However, with larger turbines, mechanical loads on the structure increase. Therefore, the actual power capture of the turbine is limited above a certain wind speed due to increasing mechanical stress on the structure and for very large wind speed, the turbine is turned off completely.

The different modes of operation of a wind turbine are defined as follows (see also Figure 1 for details):

- **Region 1** The wind speed is too low and the turbine is turned off.
- **Region 2** Below rated power, we want to maximize power capture from the wind.
- **Region 3** Above rated power, the turbine blades are pitched out off the wind such that the power capture is constant and the loads on the turbine are minimized.

The circle in Figure 1 corresponds to the so called region 2.5 (or at-rated). A turbine that is optimally sized for the site where it is installed is operating most of the time around region 2.5.

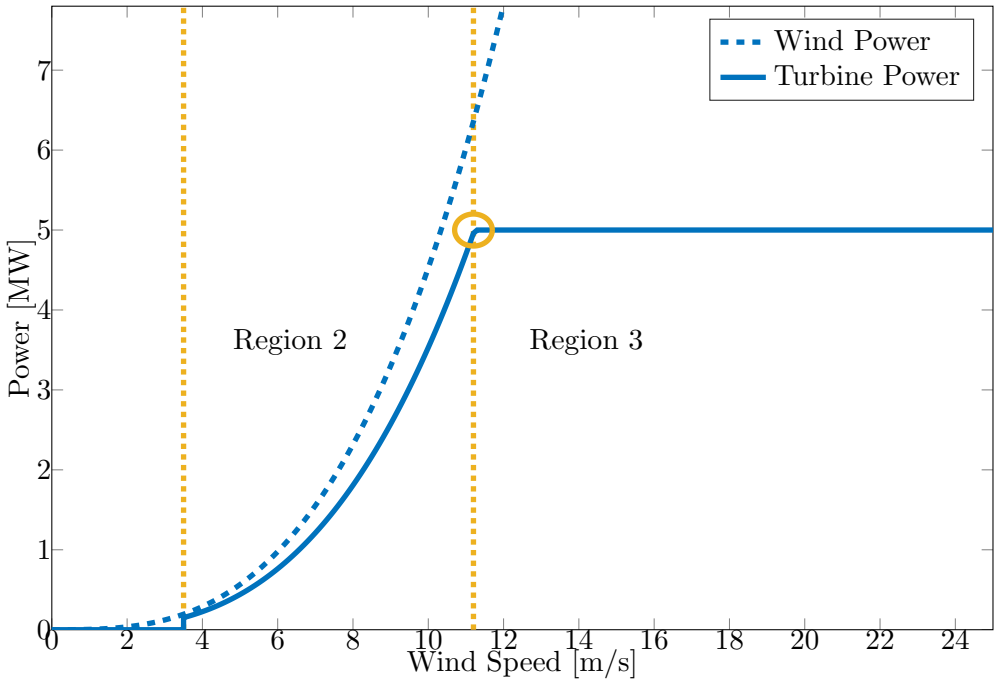

Figure 1: Different operating regions of wind turbines. Regions are separated by dashed yellow lines, the circle corresponds to region 2.5.

The control objectives for wind turbines can therefore be formulated as follows: *Maximize the power capture from the wind subject to constraints on maximum loads on the individual components of the turbine.*

2.2 Simplified nonlinear wind turbine model

We derive a simplified nonlinear model for a wind turbine that considers drive train shaft dynamics, tower fore-aft motion, and blade pitch dynamics. The model is mainly based on [35] and is compared to the model given in [9]. The wind turbine dynamics are highly nonlinear functions of the operating point defined by the tip speed ratio and the collective pitch angle. These nonlinearities are described using the aerodynamic power and thrust coefficients c_P and c_T.

Servo-elastic subsystem

In the servo elastic part, tower fore-aft bending and rotational motion are considered

$$J\dot{\Omega} + M_g/i = M_a(\dot{x}_T, \Omega, \theta, v_0) \tag{2a}$$

$$m_{Te}\ddot{x}_T + c_{Te}\dot{x}_T + k_{Te}(x_T - x_{T0}) = F_a(\dot{x}_T, \Omega, \theta, v_0). \tag{2b}$$

Equation (2a) describes the rotor dynamics with rotor speed Ω, blade pitch angle θ, tower position x_T and rotor effective wind speed v_0, where M_a is the aerodynamic torque, M_g is the generator torque, i is the gearbox ratio and J is the sum of moments of inertia at the rotational axis of hub J_H, blades J_B and generator J_G with

$$J = J_H + 3J_B + J_G/i^2.$$

Equation (2b) describes the tower fore-aft dynamics, where F_a is the aerodynamic thrust, x_{T0} is the static tower top position without aerodynamic thrust, m_{Te}, c_{Te} and k_{Te} are the tower equivalent model mass, structural damping and bending stiffness, respectively. They were calculated according to [16] as

$$\begin{aligned} m_{Te} &= 0.25m_T + m_N + m_H + 3m_B \\ c_{Te} &= 4\pi m_{Te} d_s f_0 \\ k_{Te} &= m_{Te}(2\pi f_0)^2. \end{aligned}$$

Aero-elastic subsystem

The nonlinearity of the model is contained in the aerodynamic thrust and torque acting on the rotor

$$M_a = \frac{1}{2}\rho\pi R^3 \frac{c_P(\lambda,\theta)}{\lambda} v_{rel}^2 \tag{3a}$$

$$F_a = \frac{1}{2}\rho\pi R^2 c_T(\lambda,\theta) v_{rel}^2 \tag{3b}$$

with

$$\begin{aligned} \lambda &= \frac{\Omega R}{v_{rel}} \\ v_{rel} &= (v_0 - \dot{x}_T), \end{aligned}$$

where R is the rotor radius, ρ is the air density and λ is the tip speed ratio. The tip speed ratio λ is the ratio between the tangential speed of the tip of the blade and the actual velocity of the wind. The relative wind speed v_{rel} is computed as a superposition of the tower top speed $\dot{x}_T$ and the rotor effective wind speed v_0.

Usually, the c_P and c_T coefficient are included in the model as a two-dimensional look-up table, obtained from steady state simulations using e.g. WT_Perf [32]. For this simplified nonlinear model, we approximated the c_P and c_T look-up tables by two-dimensional polynomials using a regression model.

Pitch actuator subsystem

For the pitch actuator, different levels of complexity can be considered. The most complex one is a second order lag element. However, this adds two additional states to the turbine model (six, if each blade is actuated individually). Therefore, we present also approximations of the pitch dynamics in terms of a first order lag and a time delay, which are less computational demanding. The second order lag pitch actuator dynamics is given as

$$\ddot{\theta} + 2\xi\omega\dot{\theta} + \omega^2(\theta - \theta_c) = 0, \tag{4a}$$

with θ_c the demanded pitch angle, ω the undamped natural frequency of the blade pitch actuator and ξ the damping factor of the blade pitch actuator.

Assuming reasonable fast dynamics, we can set $\ddot{\theta}$ to zero and derive the following first order lag element

$$\dot{\theta} + \frac{\omega}{2\xi}(\theta - \theta_c) = 0. \tag{4b}$$

Note, that this can also be seen as a residualization of the fastest states, such that the steady state gain matches. Finally, we can approximate the second order lag with a time delay element, such that no dynamic states are added due to the pitch actuator dynamics. The blade pitch angle is then given as

$$\theta(t) = \theta_c(t - T_p), \tag{4c}$$

where T_p is the settling time of the second order lag element computed for a settling threshold $< 2\%$ of the steady state gain. We can organize the subsystems (2), (3) and (4) in the usual nonlinear state space form

$$\dot{x} = f(x, u, d) \tag{5a}$$

$$y = h(x, u, d) \tag{5b}$$

with

$$x = [\Omega, x_T, \dot{x}_T, \theta, \dot{\theta}]^T$$
$$u = [M_g, \theta_c]^T$$
$$d = v_0$$
$$y = [\Omega, \theta]^T.$$

Depending on the used pitch actuator model (4a), the state space model can reduce to four states (4b) or three states (4c). In some setups, we could also assume that the tower accelerations $\ddot{x}_T$ is available for measurement.

2.3 Wind disturbance

The wind is acting as a disturbance on the turbine. In general, the wind field impacting the turbine is three-dimensional and stochastic. For wind turbine certification, wind fields have to be generated according to the IEC standards [1] with a certain turbulence class. For the turbine model introduced above, these wind field are reduced to one-dimensional rotor effective wind speeds v_0 acting as a scalar disturbance input to the system.

For simulation purposes, two different types of disturbance scenarios are considered: deterministic (e.g. gusts) and stochastic wind fields (with a mean wind speed and a certain turbulence intensity) affecting the rotor blades.

Comments

- In [9], mechanical losses on the shaft bearings are considered ($M_l(\Omega)$). Furthermore, first order lag dynamics are considered for the generator
$$\dot{M}_g + \frac{1}{\tau_g}(M_g - M_{g,c}) = 0.$$
- In addition to the tower fore-aft motion, the tower side-to-side motion can be added to the model.

2.4 Wind turbine controller

As discussed in section 2.1, the regions of operation of a wind turbine controller are defined based on the power constraint. *Below-rated* is defined as the region of operation when the

power constraint is not reached (region 2). Similarly, *above-rated* is the region of operation when power has to be *shed* by pitching the blades in order to limit electrical power to its constraint value (region 3). The transitioning point between below-rated and above rated is called *at-rated* (region 2.5). Region 2.5 is not just a single point (as illustrated in Figure 1) but a region to limit tip speed ratio (and hence noise emissions) at rated speed.

In variable-speed wind turbines, the conventional approach for controlling power-production relies on the design of two basic control systems: a generator-torque controller and a full-span rotor-collective blade-pitch controller. The two control systems are designed to work independently for the most part, in the below-rated and above-rated wind-speed range, respectively. The control approaches for torque and pitch discussed below are taken from the reference turbine [25], where also a more detailed description is available. A slightly more advanced control scheme for pitch and torque control loops is discussed in [8].

In the considered set up, we assume the wind turbine always to be perfectly oriented towards the the mean wind direction, i.e. yaw control is assumed to be out of scope within this effort.

Generator-Torque Controller

Below rated, the control goal is to extract maximum power from the wind. Therefore, the torque is used to control the speed to a setpoint. This setpoint is determined such that optimal tip-speed ratio (λ_{opt}) is tracked, unless the speed constraint is violated. Above rated, the generator power is held constant so that the generator torque is inversely proportional to the generator speed. The generator-torque controller given in [25] consists of a lookup table on the filtered generator speed incorporating five control regions: 1, 1.5, 2, 2.5 and 3.

In *region 1* the generator torque is zero and no power is extracted from the wind: instead, the wind is used to accelerate the rotor for start-up, i.e.

$$M_{g,d} = 0 \quad \text{for} \quad \Omega_g < \Omega_{g,1\,\max}. \tag{6a}$$

In *region 2*, the generator torque is proportional to the square of the filtered generator speed to maintain constant (optimal) tip-speed ratio (see [8] for details)

$$M_{g,d} = \frac{\pi \rho R^5 c_p}{2\lambda^3 i^3} \Omega_g^2 \quad \text{for} \quad \Omega_{g,2\,\min} < \Omega_g \leq \Omega_{g,2\,\max}. \tag{6b}$$

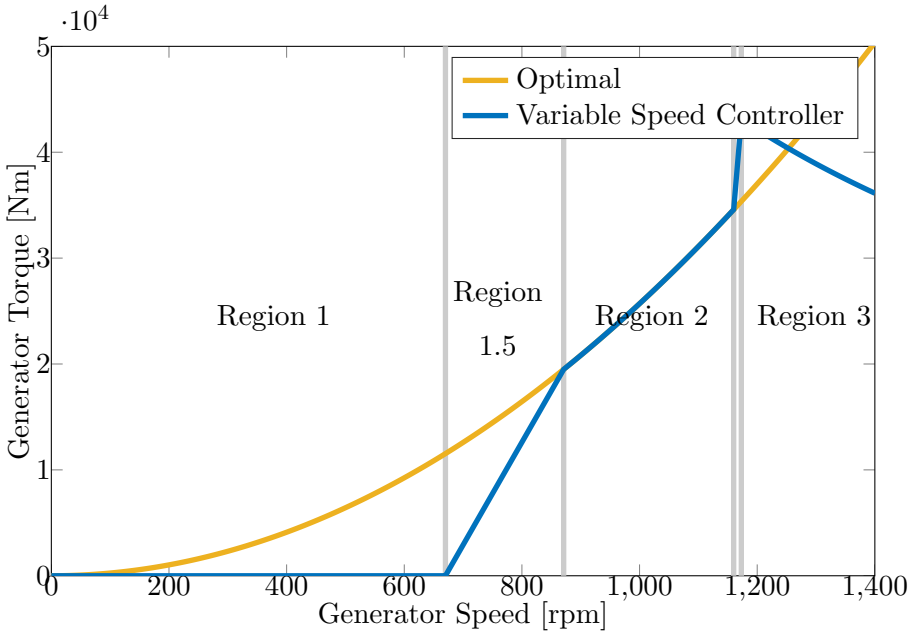

Figure 2: Torque vs. speed in the variable speed controller for the different regions. Regions are separated by gray lines (region 2.5 is not labled).

In *region 3* the generator power is held constant, so that the generator torque is inversely proportional to the filtered generator speed.

$$M_{g,d} = 1/\Omega_g \quad \text{for} \quad \Omega_g > \Omega_{g,3\,\mathrm{min}}. \tag{6c}$$

Region 1.5 is a linear transition between region 1 and region 2 and *region 2.5* a linear interpolation between region 2 and 3. The implemented variable speed controller compared to the optimal torque can be seen in Figure 2.

Collective Blade-Pitch Controller

Below region 3, the blade pitch angle is set to the optimal value θ^{opt}, corresponding to the maximum in the c_P curves. In region 3, the full-span rotor-collective blade-pitch-angle commands are computed using gain-scheduled proportional-integral (PI) control on the speed error between the filtered generator speed and the rated generator speed ($\Omega_{g,3\,\mathrm{min}}$)

$$\theta_c = K_p\Omega_g + K_I \int_0^t \Omega_g dt + K_D\dot{\Omega}_g. \tag{7}$$

The gains K_p, K_I and K_D are chosen such, that the linearized closed loop responds as a second-order system with user defined natural frequency ω_{cl} and damping ratio ζ_{cl}. In the considered setup, the derivative term K_D is neglected, and only a PI-controller is implemented. With the above specifications and (2a), the controller coefficients can be chosen as

$$K_p = \frac{2J\Omega\zeta_{cl}\omega_{cl}}{i\left(-\frac{\partial P}{\partial \theta}\right)},$$
$$K_i = \frac{J\Omega\omega_{cl}^2}{i\left(-\frac{\partial P}{\partial \theta}\right)}.$$

The blade sensitivity $\partial P/\partial\theta$ is an aerodynamic property of the rotor that depends on the wind speed, rotor speed and blade pitch angle. It can be computed by linearization analysis for different wind speeds at rated rotor speed such that the blade angle produces the rated mechanical power. As a result, the pitch sensitivity varies nearly linearily with the blade pitch angle. Therefore, the gains of the PID controller can be gain-scheduled by the blade pitch angle, i.e.

$$K_p(\theta) = \frac{2J\Omega\zeta_{cl}\omega_{cl}}{i\left(-\frac{\partial P}{\partial \theta}|_{\theta=0}\right)} f(\theta),$$
$$K_i(\theta) = \frac{J\Omega\omega_{cl}^2}{i\left(-\frac{\partial P}{\partial \theta}|_{\theta=0}\right)} f(\theta),$$

where $f(\theta)$ is the gain scheduling factor.

Constraints on control action

We consider saturation for the pitch and torque signal as well as constraints on the rate of change for these signals. Especially the pitch rate limitations usually limit the performance of the controller.

Additional control objectives

The above described controller is the most basic turbine controller. Additional control objectives considered in conventional wind turbine controllers are

- tower damping to reduce tower fore-aft and tower side-to-side fatigue loads,
- drive train damping to attenuate vibrations in the gear box,
- individual pitch control to generate independent pitch commands for all blades to compensate asymmetric blade loads.

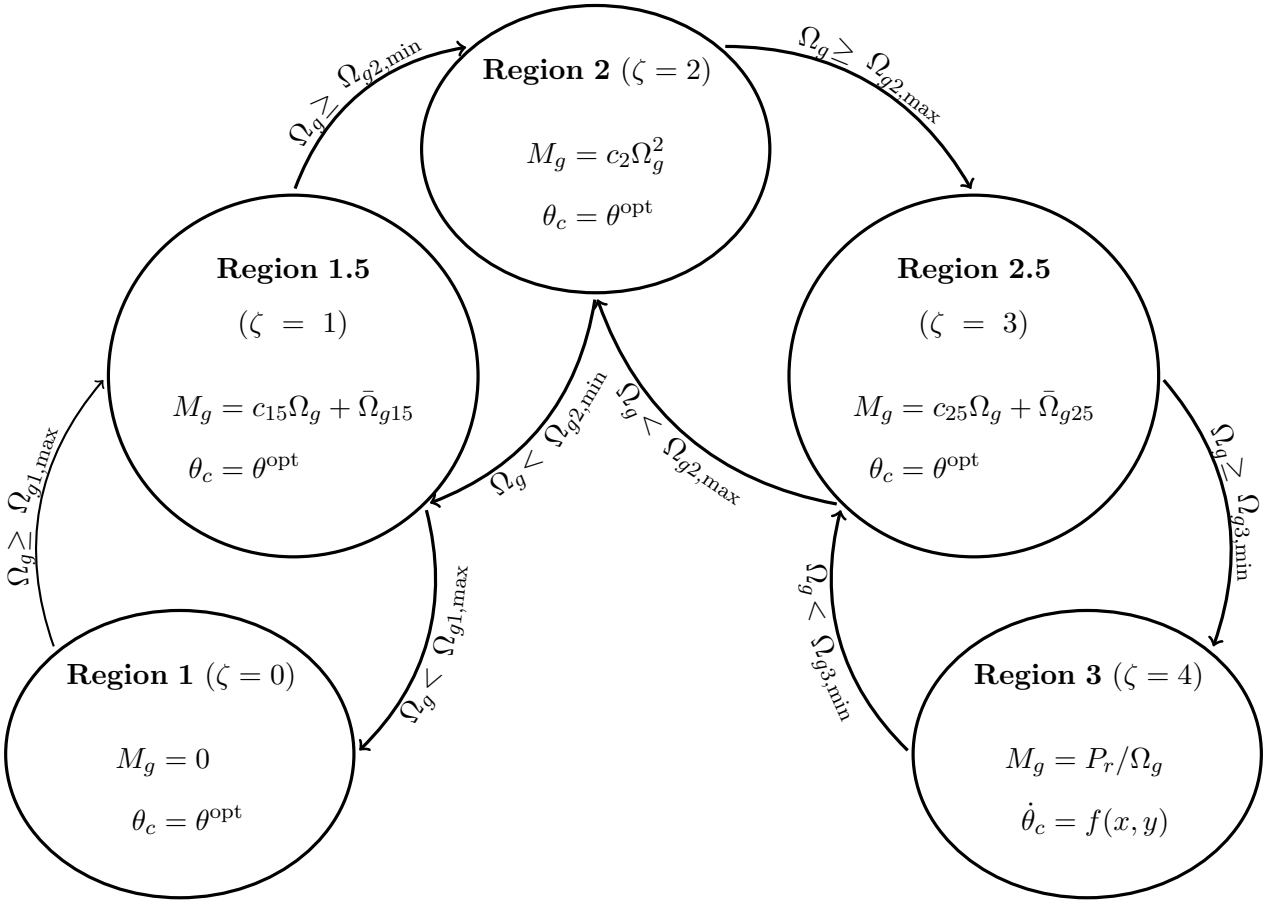

Figure 3: Wind turbine as a hybrid automaton.

2.5 Combined hybrid model

As the system described in the previous sections includes continuous-time dynamics and discrete-time events, we can rewrite the system as an impulsive system, using the formalism of [17]. The system composed of the joint continuous and discrete state $q = [x_{cl}^T, \zeta]^T$ as given in (5), (6) and (7) can be written as follows

$$\frac{d}{dt}\begin{bmatrix} x_{cl} \\ \zeta \end{bmatrix} = \begin{bmatrix} F(x_{cl}) \\ 0 \end{bmatrix}, \quad \text{for } q \in C \tag{8a}$$

$$\begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 1 \end{bmatrix} \quad \text{for } q \in D_1, \qquad \begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 0 \end{bmatrix} \quad \text{for } q \in D_2$$

$$\begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 2 \end{bmatrix} \quad \text{for } q \in D_3, \qquad \begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 1 \end{bmatrix} \quad \text{for } q \in D_4 \tag{8b}$$

$$\begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 3 \end{bmatrix} \quad \text{for } q \in D_5, \qquad \begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 2 \end{bmatrix} \quad \text{for } q \in D_6$$

$$\begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 4 \end{bmatrix} \quad \text{for } q \in D_7, \qquad \begin{bmatrix} x_{cl}^+ \\ \zeta^+ \end{bmatrix} = \begin{bmatrix} x_{cl} \\ 3 \end{bmatrix} \quad \text{for } q \in D_8$$

$$\begin{aligned}
D_1 &: \{\zeta = 0; \quad \Omega_g \geq \Omega_{g1,\max}\} \\
D_2 &: \{\zeta = 1; \quad \Omega_g < \Omega_{g1,\max}\} \\
D_3 &: \{\zeta = 1; \quad \Omega_g \geq \Omega_{g2,\min}\} \\
D_4 &: \{\zeta = 2; \quad \Omega_g < \Omega_{g2,\min}\} \\
D_5 &: \{\zeta = 2; \quad \Omega_g \geq \Omega_{g2,\max}\} \\
D_6 &: \{\zeta = 3; \quad \Omega_g < \Omega_{g2,\max}\} \\
D_7 &: \{\zeta = 3; \quad \Omega_g \geq \Omega_{g3,\min}\} \\
D_8 &: \{\zeta = 4; \quad \Omega_g < \Omega_{g3,\min}\},
\end{aligned}$$

where $C \in \mathbb{R}^n$ defines the flow set and, in virtue of Figure 3, the jump set D can be defined as $D = \bigcup_{i=1}^{8} D_i$.

2.6 Power capture and mechanical loads

The electrical power P_{el} is calculated by

$$P_{el} = \eta M_g \Omega / i, \tag{9}$$

where η represents the efficiency of the electro-mechanical energy conversion.

Due to the flexible structure, mechanical loads are an important driving factor for the controller design of wind turbines. For the previously defined turbine model, only the tower base fore-aft bending moment M_{yT} can be considered

$$M_{yT} = h_H(c_T \dot{x}_T + k_T x_T). \tag{10}$$

Usually, in addition to the dynamic loads, damage equivalent loads (DELs) are also calculated for the turbine. The basic idea is a reduction of the complicated load distributions over the whole lifetime of turbine to a single load indicator using the rain flow counting algorithm [4] together with a Weibull distribution. The DEL is the amplitude of the reference loading cycle causing in theory the same damage as the corresponding load cycle distribution.

2.7 High fidelity model

Typically, wind turbines are simulated using high fidelity simulation environments as e.g. FAST (Fatigue, Aerodynamics, Structures and Turbulence) [24]. The FAST model employs a combined modal and multibody dynamics formulation. The model for a three bladed horizontal axis turbine contains up to 24 DOFs. Additionally, complex modelling of the aerodynamic is included. For the purpose of this project, we work on developing a model that reflects the behaviour of a wind turbine similar in a high fidelity simulation environment that can be used for verification and validation purposes.

3 Model of a variable-size smart grid

This section is structured in two parts: modeling of the smart grid components (Subsection 3.1) and definition of possible smart grid configurations and related compositional models (Subsection 3.2).

More precisely, we start describing the models of the components of a smart-grid, focusing, in particular, on the cooling of a district network, which is composed of multiple buildings,

possibly sharing resources such as storages, chillers, combined heat and power units, and renewable power generators like wind turbines. We adopt a modular and control perspective oriented to the energy management of the smart grid along some finite time horizon $[t_i, t_f]$. To this purpose, the time horizon is divided into M time slots of duration Δt and the contribution in terms of energy requested/provided by the different components per time slot along the discretized control horizon is provided. The smart grid components can then be combined via energy balance equations. Some of the inputs are left free and can be set, e.g., to minimize the electrical energy costs through some optimal control strategy. Control design can be performed in a distributed way to preserve privacy information related to each building consumption, and/or to reduce the computation load and the amount of information transmitted.

Depending on the components of the smart grid and their description, the resulting overall model is a continuous system with continuous state and input variables, convex in the (continuous) control input, or hybrid with state and input that have both a continuous and a discrete component and a dynamics that is affine in the continuous control input for each discrete mode. The piecewise affine dependence on the continuous control input, jointly with the presence of discrete inputs, makes the design of an optimal control strategy amenable for a mixed integer linear programming formulation.

Disturbance inputs are also accounted for when modeling the various components. The presence of disturbances calls for some degree of robustness in control design. In the case when the disturbance is stochastic, one can e.g. formulate a chance-constrained optimization problem with an average cost function to minimize subject to probabilistic constraints. The presence of disturbances makes also the verification of the system performance more challenging.

3.1 Components of the smart grid

3.1.1 Building

Consider a building composed of n_z zones, each one with its own temperature profile $T_{z,j}$, $j = 1, \ldots, n_z$, to track. We can then define the temperature profile of the building as the following vector

$$\boldsymbol{T_z} = [T_{z,1} \cdots T_{z,n_z}]^\top.$$

The cooling energy demand of the building for tracking the temperature profile $\boldsymbol{T_z}$ during the discretized time horizon $[t_i, t_f]$ is given by

$$J = \sum_{k=1}^{M} E_{r,c}(k),$$

where $E_{r,c}(k)$ is the cooling energy request during the time slot k, to be summed up over all the M time slots. Note that the subscript r, c in $E_{r,c}(k)$ stands for request (r) cooling (c). In turn, we can write:

$$E_{r,c}(k) = \sum_{j=1}^{n_z} E_{c,j}(k), \tag{11}$$

where $E_{c,j}(k)$ is the energy request of zone j during slot k and we are adding up the contributions over all n_z zones composing the building. Evidently, the cooling energy request of zone j depends on its temperature profile $T_{z,j}$ and we shall now make this dependence explicit by describing the contributions to $E_{c,j}(k)$. $E_{c,j}(k)$ is the sum of four contributions, namely

$$E_{c,j}(k) = E_{w,j}(k) + E_{p,j}(k) + E_{\text{int},j}(k) + E_{z,j}(k). \tag{12}$$

$E_{w,j}$ in (12) is the amount of energy exchanged between the walls and zone j, $E_{p,j}$ and $E_{\text{int},j}$ is the heat produced respectively by people and by other sources of heat inside zone j, and $E_{z,j}$ is the energy contribution of the thermal inertia of zone j.

We shall show next that the energy contributions in (12) are affine as a function of $\boldsymbol{T_z}$. This is particularly convenient if $\boldsymbol{T_z}$ is taken as control input to be set, e.g., for energy cost minimization.

Wall-zone energy exchange E_w. The building is composed of zones that are separated through walls one from the other, and from the outside ambient as well. Walls can be modeled as the composition of one-dimensional finite volumes. Each wall is divided into vertical layers ('slices') that differ in width and material composition. The area of each slice coincides with the wall area and each slice is assumed to have a uniform density and a uniform temperature. The one-dimensional discretization has been chosen exploiting the fact that the heat flow is perpendicular to the surface it is passing through. Each internal slice exchanges heat only with nearby slices through conduction, whilst boundary slices also exchange heat via convection and thermal radiation through surfaces that are exposed towards either a zone or the outside of the building. External surfaces are assumed to be grey and opaque, with

equal absorbance and emissivity and with zero transmittance. Absorbance and emissivity are wavelength-dependent quantities, and here we shall consider two different values for shortwave and longwave radiation, [30].

The energy balance equation for the i^{th} slice is given by:

$$\begin{aligned}\dot{T}_i =&\frac{1}{C_i}\Big[(k_i^{i-1}+h_i^{i-1})T_{i-1}+(k_i^{i+1}+h_i^{i+1})T_{i+1}-(k_i^{i-1}+h_i^{i-1}+k_i^{i+1}+h_i^{i+1})T_i\\&+\alpha_i^S Q^S+\alpha_i^L Q^L-\varepsilon_i Q_r(T_i)+Q_{g,i}\Big],\end{aligned}\tag{13}$$

where T_i denotes the temperature of the slice, C_i being its thermal capacity per unit area, and k_i^j and h_i^j, with $j=i\pm1$, representing respectively the conductive and convective heat transfer coefficients between the i^{th} and the j^{th} slice. The incoming shortwave and longwave radiation power per unit area are Q^S and Q^L, respectively, and α_i^S and α_i^L are the corresponding absorbance rates. $Q_r(T_i)$ is the emitted radiation, $\varepsilon_i<1$ is the emissivity and $Q_{g,i}$ is the thermal power generation inside slice i. In (13), $i=1,\ldots,m$, m being the number of slices composing the wall and being superscripts/subscripts 0 and $m+1$ denote either a zone of the building or the outside of the building. Simply set $k_1^0=k_m^{m+1}=0$ if there is no thermal conduction on walls boundary surfaces, and set

$$\begin{aligned}&h_i^{i-1}=0,\ i>1\\&h_i^{i+1}=0,\ i<m\\&\alpha_i^S=\alpha_i^L=\varepsilon_i=0,\ 1<i<m\end{aligned}$$

since there is no thermal convection nor radiation in between slices. Given that each wall is assumed to be a gray body, the power $Q_r(T_i)$ radiated from each slice is governed by $Q_r(T_i)=\sigma T_i^4$, where σ is the Stefan-Boltzmann constant. This expression is approximately linear around the slice mean operating temperature $\bar{T}_i$ so that it can be replaced by

$$Q_r(T_i)=4\sigma\bar{T}_i^3T_i-3\sigma\bar{T}_i^4.\tag{14}$$

If we consider a generic wall w composed of m slices, the evolution of $\boldsymbol{T_w}=[T_{w,1}\cdots T_{w,m}]^\top$, with $T_{w,i}$ denoting the temperature of the i^{th} slice of wall w, can be described in matrix form by

$$\dot{\boldsymbol{T}}_{\boldsymbol{w}}=\boldsymbol{A_w}\boldsymbol{T_w}+\boldsymbol{B_w}\boldsymbol{T_z}+\boldsymbol{W_w}\boldsymbol{d},\tag{15}$$

where we recall that $\boldsymbol{T_z}$ is the vector containing the temperature of the n_z zones, that can be treated as control inputs. The disturbance

$$\boldsymbol{d} = [T_{out}\ Q^S\ Q^L\ 1]^\top$$

collects the outdoor temperature T_{out}, and the incoming shortwave Q^S and longwave Q^L radiation. The constant 1 in $\boldsymbol{d}$ is introduced to account for the constant term in (14). Finally, $\boldsymbol{A_w}$, $\boldsymbol{B_w}$ and $\boldsymbol{W_w}$ are suitably defined matrices that are easily derived based on the scalar equation (13), whose coefficients depend on the wall characteristics.

Equation (15) refers to a single wall. If there are n_w walls in the building, then, we can collect all wall temperatures in a vector:

$$\boldsymbol{T} = [\boldsymbol{T_1}^\top \cdots \boldsymbol{T_{n_w}}^\top]^\top$$

and write the following equation for the evolution in time of $\boldsymbol{T}$:

$$\dot{\boldsymbol{T}} = \boldsymbol{A}\boldsymbol{T} + \boldsymbol{B}\boldsymbol{T_z} + \boldsymbol{W}\boldsymbol{d}, \tag{16}$$

where $\boldsymbol{A}$ is a block-diagonal matrix in which the w^{th} block is $\boldsymbol{A_w}$, $\boldsymbol{B} = [\boldsymbol{B_1}^\top \cdots \boldsymbol{B_{n_w}}^\top]^\top$, and $\boldsymbol{W} = [\boldsymbol{W_1}^\top \cdots \boldsymbol{W_{n_w}}^\top]^\top$.

As for the amount of heat exchanged between each wall and each adjacent zone, the thermal power transferred from wall w with surface S_w to zone j is given by

$$Q_{w\to j} = S_w h_{w,b}^{b'}(T_{w,b} - T_{z,j}),$$

$w = 1, \dots, n_w$ and $j = 1, \dots, n_z$. The pair (b, b') can either be $(1, 0)$ or $(m, m+1)$ according to the notation introduced for (13). The total amount of thermal power transferred from the building walls to zone j can be expressed as

$$Q_{b,j} = \sum_{w \in \mathcal{W}_j} Q_{w\to j}$$

where $\mathcal{W}_j$ is the set of walls w adjacent to zone j. Defining $\boldsymbol{Q} = [Q_{b,1} \cdots Q_{b,n_z}]^\top$, we obtain

$$\boldsymbol{Q} = \boldsymbol{C}\boldsymbol{T} + \boldsymbol{D}\boldsymbol{T_z}, \tag{17}$$

where $\boldsymbol{C}$ and $\boldsymbol{D}$ are suitable matrices. From (16) and (17), we finally get

$$\begin{cases} \dot{\boldsymbol{T}} = \boldsymbol{A}\boldsymbol{T} + \boldsymbol{B}\boldsymbol{T_z} + \boldsymbol{W}\boldsymbol{d} \\ \boldsymbol{Q} = \boldsymbol{C}\boldsymbol{T} + \boldsymbol{D}\boldsymbol{T_z} \end{cases} \tag{18}$$

Note that the obtained model, though linear, can be quite large. However, following [29], its order can be greatly reduced by applying the model reduction algorithm based on Hankel's Single Value Decomposition (HSVD), which for a given transfer matrix $W(s)$ of McMillan order n finds the $\hat{W}(s)$ of order $r < n$ such that the Hankel norm $||W(s) - \hat{W}(s)||_H$ is minimized.

For ease of computation, we consider a discretized version of (18). Given the linearity of (18), it holds that

$$\begin{cases} \boldsymbol{T_{k+1}} = e^{\boldsymbol{A}\Delta t}\boldsymbol{T_k} + \int_{k\Delta t}^{(k+1)\Delta t} e^{\boldsymbol{A}((k+1)\Delta t-\tau)}(\boldsymbol{B}\boldsymbol{T_z}(\tau) + \boldsymbol{W}\boldsymbol{d}(\tau))d\tau, \\ \boldsymbol{Q}_k = \ \boldsymbol{C}\boldsymbol{T_k} + \boldsymbol{D}\boldsymbol{T_z}(k\Delta t) \end{cases} \tag{19}$$

where $\boldsymbol{T_k} = \boldsymbol{T}(k\Delta t)$ and $\boldsymbol{Q_k} = \boldsymbol{Q}(k\Delta t)$, $k = 1, \ldots, M$.

If we assume that $\boldsymbol{T_z}$ and $\boldsymbol{d}$ are linearly varying within each time slot, i.e.,

$$\begin{aligned} \boldsymbol{T_z}(\tau) &= \frac{\boldsymbol{T_{z,k+1}} - \boldsymbol{T_{z,k}}}{\Delta t}(\tau - k\Delta t) + \boldsymbol{T_{z,k}} \\ \boldsymbol{d}(\tau) &= \frac{\boldsymbol{d_{k+1}} - \boldsymbol{d_k}}{\Delta t}(\tau - k\Delta t) + \boldsymbol{d_k}, \end{aligned} \tag{20}$$

$\tau \in [k\Delta t, (k+1)\Delta t)$, $k = 1, \ldots, M$, where $\boldsymbol{T_{z,k}} = \boldsymbol{T_z}(k\Delta t)$ and $\boldsymbol{d_k} = \boldsymbol{d}(k\Delta t)$, then the integral in (19) can be computed analytically and the discretized system can be expressed as follows

$$\begin{cases} \boldsymbol{T_{k+1}} = \boldsymbol{\Gamma_x}\boldsymbol{T_k} + \boldsymbol{\Gamma_{u,1}}\boldsymbol{T_{z,k+1}} + (\boldsymbol{\Gamma_{u,0}} - \boldsymbol{\Gamma_{u,1}})\boldsymbol{T_{z,k}} + \boldsymbol{\Gamma_{w,1}}\boldsymbol{d_{k+1}} + (\boldsymbol{\Gamma_{w,0}} - \boldsymbol{\Gamma_{w,1}})\boldsymbol{d_k} \\ \boldsymbol{Q}_k = \ \boldsymbol{C}\boldsymbol{T_k} + \boldsymbol{D}\boldsymbol{T_{z,k}} \end{cases}$$

where we set

$$\begin{aligned} \boldsymbol{\Gamma_x} &= e^{\boldsymbol{A}\Delta t} \\ \boldsymbol{\Gamma_{u,1}} &= \frac{1}{\Delta t}\left(\int_0^{\Delta t} e^{\boldsymbol{A}s}(\Delta t - s)ds\right)\boldsymbol{B} \\ \boldsymbol{\Gamma_{u,0}} &= \left(\int_0^{\Delta t} e^{\boldsymbol{A}s}ds\right)\boldsymbol{B} \\ \boldsymbol{\Gamma_{w,1}} &= \frac{1}{\Delta t}\left(\int_0^{\Delta t} e^{\boldsymbol{A}s}(\Delta t - s)ds\right)\boldsymbol{W} \\ \boldsymbol{\Gamma_{w,0}} &= \left(\int_0^{\Delta t} e^{\boldsymbol{A}s}ds\right)\boldsymbol{W}. \end{aligned}$$

Applying the transformation

$$\boldsymbol{\xi_k} = \boldsymbol{T_k} - \boldsymbol{\Gamma_{u,1}}\boldsymbol{T_{z,k}} - \boldsymbol{\Gamma_{w,1}}\boldsymbol{d_k}$$

as described in [40], we finally obtain

$$\begin{cases} \boldsymbol{\xi_{k+1}} = \boldsymbol{\Gamma_x}\boldsymbol{\xi_k} + ((\boldsymbol{\Gamma_x} - \boldsymbol{I})\boldsymbol{\Gamma_{u,1}} + \boldsymbol{\Gamma_{u,0}})\boldsymbol{T_{z,k}} + ((\boldsymbol{\Gamma_x} - \boldsymbol{I})\boldsymbol{\Gamma_{w,1}} + \boldsymbol{\Gamma_{w,0}})\boldsymbol{d_k} \\ \boldsymbol{Q_k} = \ \boldsymbol{C}\boldsymbol{\xi_k} + (\boldsymbol{C}\boldsymbol{\Gamma_{u,1}} + \boldsymbol{D})\boldsymbol{T_{z,k}} + \boldsymbol{C}\boldsymbol{\Gamma_{w,1}}\boldsymbol{d_k} \end{cases},$$

which, by dropping the bold notation for vectors and matrices, can be rewritten as the following discrete-time system

$$\begin{cases} x(k+1) = \tilde{A}x(k) + \tilde{B}u(k) + \tilde{W}w(k) \\ y(k) = \tilde{C}x(k) + \tilde{D}u(k) + \tilde{V}w(k) \end{cases}, \tag{21}$$

with state, control input, disturbance, and output

$$x(k) = \boldsymbol{\xi_k},\ u(k) = \boldsymbol{T_{z,k}},\ w(k) = \boldsymbol{d_k},\ y(k) = \boldsymbol{Q_k},$$

and matrices

$$\begin{aligned} \tilde{A} &= \boldsymbol{\Gamma_x} & \tilde{B} &= (\boldsymbol{\Gamma_x} - \boldsymbol{I})\boldsymbol{\Gamma_{u,1}} + \boldsymbol{\Gamma_{u,0}} \\ \tilde{W} &= (\boldsymbol{\Gamma_x} - \boldsymbol{I})\boldsymbol{\Gamma_{w,1}} + \boldsymbol{\Gamma_{w,0}} & \tilde{C} &= \boldsymbol{C} \\ \tilde{D} &= \boldsymbol{C\Gamma_{u,1}} + \boldsymbol{D} & \tilde{V} &= \boldsymbol{C\Gamma_{w,1}}. \end{aligned}$$

From (21) one can derive the expression of $x(k)$ and $y(k)$ as a function of the initial state and the control input and disturbance up to k:

$$\begin{aligned} x(k) &= \tilde{A}^k x(0) + \sum_{h=0}^{k-1} \tilde{A}^{k-1-h} \left(\tilde{B}u(h) + \tilde{W}w(h) \right) \\ y(k) &= \tilde{C}x(k) + \tilde{D}u(k) + \tilde{V}w(k) \end{aligned}.$$

Setting

$$\begin{aligned} \boldsymbol{u} &= [u^\top(0) \cdots u^\top(M)]^\top \\ \boldsymbol{w} &= [w^\top(0) \cdots w^\top(M)]^\top \\ \boldsymbol{y} &= [y^\top(0) \cdots y^\top(M)]^\top \end{aligned} \tag{22}$$

we finally have that

$$\boldsymbol{y} = Fx(0) + G\boldsymbol{u} + H\boldsymbol{w} \tag{23}$$

where F, G and H are suitably defined matrices.

Recalling that $y(k) = \boldsymbol{Q_k}$ and assuming that each component of $\boldsymbol{Q}(t)$ varies linearly within each time slot, vector $E_w(k) = [E_{w,1}(k) \cdots E_{w,n_z}(k)]^\top$ of the overall thermal energy transferred from the walls to each zone can be computed as

$$E_w(k) = \frac{\Delta t}{2}(\boldsymbol{Q_{k-1}} + \boldsymbol{Q_k}) = \frac{\Delta t}{2}(y(k-1) + y(k)), \tag{24}$$

$k = 1, \dots, M$. Finally, defining $\boldsymbol{E_w} = [E_w^\top(1) \cdots E_w^\top(M)]^\top$, from (23) and (24), we can derive the expression for the thermal energy exchanged by the building structure to the each thermal zone:

$$\boldsymbol{E_w} = \tilde{F}x(0) + \tilde{G}\boldsymbol{u} + \tilde{H}\boldsymbol{w},$$

where matrices $\tilde{F}$, $\tilde{G}$, $\tilde{H}$ are given by

$$\tilde{F} = \frac{\Delta t}{2}\begin{bmatrix} \tilde{C}+\tilde{C}\tilde{A} & \tilde{D}+\tilde{C}\tilde{B} & \tilde{V}+\tilde{C}\tilde{W} \\ \tilde{C}\tilde{A}+\tilde{C}\tilde{A}^2 & \tilde{C}\tilde{B}+\tilde{C}\tilde{A}\tilde{B} & \tilde{C}\tilde{W}+\tilde{C}\tilde{A}\tilde{W} \\ \tilde{C}\tilde{A}^2+\tilde{C}\tilde{A}^3 & \tilde{C}\tilde{A}\tilde{B}+\tilde{C}\tilde{A}^2\tilde{B} & \tilde{C}\tilde{A}\tilde{W}+\tilde{C}\tilde{A}^2\tilde{W} \\ \vdots & \vdots & \vdots \\ \tilde{C}\tilde{A}^{M-1}+\tilde{C}\tilde{A}^M & \tilde{C}\tilde{A}^{M-2}\tilde{B}+\tilde{C}\tilde{A}^{M-1}\tilde{B} & \tilde{C}\tilde{A}^{M-2}\tilde{W}+\tilde{C}\tilde{A}^{M-1}\tilde{W} \end{bmatrix}$$

$$\tilde{G} = \frac{\Delta t}{2}\begin{bmatrix} \tilde{D} & 0 & \cdots & 0 \\ \tilde{D}+\tilde{C}\tilde{B} & \tilde{D} & \cdots & 0 \\ \tilde{C}\tilde{B}+\tilde{C}\tilde{A}\tilde{B} & \tilde{D}+\tilde{C}\tilde{B} & \cdots & 0 \\ \tilde{C}\tilde{A}\tilde{B}+\tilde{C}\tilde{A}^2\tilde{B} & \tilde{C}\tilde{B}+\tilde{C}\tilde{A}\tilde{B} & \cdots & 0 \\ \vdots & \vdots & \cdot & \vdots \\ \tilde{C}\tilde{A}^{M-3}\tilde{B}+\tilde{C}\tilde{A}^{M-2}\tilde{B} & \tilde{C}\tilde{A}^{M-4}\tilde{B}+\tilde{C}\tilde{A}^{M-3}\tilde{B} & \cdots & \tilde{D} \end{bmatrix}$$

$$\tilde{H} = \frac{\Delta t}{2}\begin{bmatrix} \tilde{V} & 0 & \cdots & 0 \\ \tilde{V}+\tilde{C}\tilde{W} & \tilde{V} & \cdots & 0 \\ \tilde{C}\tilde{W}+\tilde{C}\tilde{A}\tilde{W} & \tilde{V}+\tilde{C}\tilde{W} & \cdots & 0 \\ \tilde{C}\tilde{A}\tilde{W}+\tilde{C}\tilde{A}^2\tilde{W} & \tilde{C}\tilde{W}+\tilde{C}\tilde{A}\tilde{W} & \cdots & 0 \\ \vdots & \vdots & \cdot & \vdots \\ \tilde{C}\tilde{A}^{M-3}\tilde{W}+\tilde{C}\tilde{A}^{M-2}\tilde{W} & \tilde{C}\tilde{A}^{M-4}\tilde{W}+\tilde{C}\tilde{A}^{M-3}\tilde{W} & \cdots & \tilde{V} \end{bmatrix}$$

People energy contribution E_p. Occupancy implies heat production and, in crowded places (e.g. offices) it is an absolutely relevant contribution to heat generation. According to an empirical model documented in [2], the Basal Metabolic Rate (BMR) quantifies the heat production rate of a person in a thermoneutral environment ($23°C$), mentally and physically at rest, after more than 12 hours after the last meal. The BMR depends on different factors, but we are here mainly concerned in describing how it relates with the ambient temperature. The heat rate $Q_{p,j}(t)$ produced by a number of occupants $n_p(t)$ inside zone j at temperature

T_z is given by

$$Q_{p,j}(t) = n_p(t)(p_2 T_{z,j}^2(t) + p_1 T_{z,j}(t) + p_0), \tag{25}$$

where p_0, p_1, and p_2 are suitable coefficients.

Although expression (25) is not convex as a function of $T_{z,j}$, it is almost linear in a sensible operating temperature range and can thus be linearized around some comfort temperature $\bar{T}_{z,j}$, thus obtaining:

$$\begin{aligned} Q_{p,j}(t) &= n_p(t)\Big((2p_2\bar{T}_{z,j} + p_1)(T_{z,j}(t) - \bar{T}_{z,j}) + p_2\bar{T}_{z,j}^2 + p_1\bar{T}_{z,j} + p_0\Big) \\ &= n_p(t)(\tilde{p}_1 T_{z,j}(t) + \tilde{p}_0). \end{aligned} \tag{26}$$

Recalling that $T_{z,j}(t)$ is assumed to be linear within each time slot (see (20)), if we take $n_p(t)$ as a linear function of time as suggested in [7], then equation (26) can be analytically integrated from $(k-1)\Delta t$ to $k\Delta t$ to obtain the energy transferred to zone j

$$E_{p,j}(k) = q_{2,k}(n_p)T_{z,j}(k\Delta t) + q_{1,k}(n_p)T_{z,j}((k-1)\Delta t) + q_{0,k}(n_p),$$

where

$$\begin{aligned} q_{2,k}(n_p) &= \tilde{p}_1\left(\frac{1}{3}n_{p,k} + \frac{1}{6}n_{p,k-1}\right)\Delta t \\ q_{1,k}(n_p) &= \tilde{p}_1\left(\frac{1}{6}n_{p,k} + \frac{1}{3}n_{p,k-1}\right)\Delta t \\ q_{0,k}(n_p) &= \tilde{p}_0\left(\frac{1}{2}n_{p,k} + \frac{1}{2}n_{p,k-1}\right)\Delta t, \end{aligned} \tag{27}$$

with $n_{p,k} = n_p(k\Delta t)$.

The total amount of energy transferred to all zones in each time slot can be collected in a vector $E_p(k) = [E_{p,1}(k)\cdots E_{p,n_z}(k)]^\top$ and then, defining $\boldsymbol{E_p} = [E_p^\top(1)\cdots E_p^\top(M)]^\top$, one can write that

$$\boldsymbol{E_p} = N(n_p)\boldsymbol{u} + e(n_p),$$

where $N(n_p)$ and $e(n_p)$ depend on the coefficients (27) and $\mathbf{u}$ is defined in (22).

Other internal energy contributions E_{int}. There are many other types of heat sources that may affect the internal energy of a building, e.g. internal lighting, electrical equipment, daylight radiation through windows, etc. The overall heat flow rate transferred to zone j can be expressed as the sum of three contributions, namely

$$Q_{\text{int},j}(t) = \alpha_j(t)Q^S(t) + \kappa_j I_{\mathbf{R}^+}(n_p(t)) + \lambda_j, \tag{28}$$

where $\alpha_j(t)$ is a coefficient that takes into account the mean absorbance coefficient of zone j, the transmittance coefficients of the windows and their areas, sun view and shading factors, and radiation incidence angle. The thermal energy contribution to zone j due to internal lightening and electrical equipment is composed of two contributions: a constant term λ_j, and an additional therm κ_j that represents the change in internal lightening and electrical equipment when people are present. $I_{\mathbf{R}^+}(\cdot)$ denote the indicator function on the positive real values. Note that $Q_{\text{int},j}$ does not depend on the longwave radiation because windows are usually shielded against it. Similarly to the previous section, (28) can be discretized and integrated in order to obtain the energy $E_{\text{int},j}(k)$ during slot k. We can collect the thermal energies of the zones in a vector $E_{\text{int}}(k) = [E_{\text{int},1}(k) \cdots E_{\text{int},n_z}(k)]^\top$, and, finally, define

$$\boldsymbol{E_{\text{int}}} = [E_{\text{int}}^\top(1) \cdots E_{\text{int}}^\top(M)]^\top.$$

Zones energy contributions E_z. Observe that in order to lower the temperature of a zone we need to draw energy from the zone itself. This contribution to the overall thermal energy (12) in the building can be expressed as

$$E_{z,j}(k) = -C_{z,j}(T_{z,j}(k\Delta t) - T_{z,j}((k-1)\Delta t)), \tag{29}$$

where $C_{z,j}$ is the heat capacity of the j^{th} zone. To account for all time slots and all zones in the building, we can use (29) and derive the following equation in matrix form:

$$\boldsymbol{E_z} = Z\boldsymbol{u} + z,$$

where

$$\boldsymbol{E_z} = [E_z^\top(1) \cdots E_z^\top(M)]^\top$$

with

$$E_z(k) = [E_{z,1}(k) \cdots E_{z,n_z}(k)]^\top$$

and

$$Z = \begin{bmatrix} -C_{z,j} & 0 & & \\ C_{z,j} & -C_{z,j} & 0 & \\ 0 & C_{z,j} & -C_{z,j} & 0 \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix} \quad z = \begin{bmatrix} C_{z,j}T_{z,j}(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

3.1.2 Chiller

A chiller plant converts electrical energy into cooling energy. The cooling energy is then transferred to the building via, e.g., the chilled water circuit. Chillers can be modeled through the equation:

$$E_{ch,l} = \frac{a_1 T_o T_{cw} \Delta t + a_2 (T_o - T_{cw}) \Delta t + a_4 T_o E_{ch,c}}{T_{cw} - \frac{a_3}{\Delta t} E_{ch,c}} - E_{ch,c}, \tag{30}$$

where $E_{ch,l}$ is the electrical energy absorbed by the chiller in order to provide the cooling energy $E_{ch,c}$ in a time slot of duration Δt. Note that $E_{ch,l}$ depends also on the outdoor temperature T_o and the temperature of the cooling water T_{cw}. The latter is typically regulated by low level controllers so that it is maintained almost at some prescribed fixed value. The chiller description (30) is derived from the original Ng-Gordon model [18] which is based on entropy and energy balance equations. Coefficients a_1, a_2, a_3, a_4 characterize the chiller performance. Depending on their values, we can have different efficiency curves as given by the so-called coefficient of performance (COP). The COP is the ratio between the produced cooling energy and the corresponding electrical energy consumption. As such, the larger is the COP, the more efficient is the chiller. Figure 4 shows some COP curves for chillers of different size, whose coefficients are listed in Table 1, [7]. Sensible values were chosen for the temperatures T_o and T_{cw}.

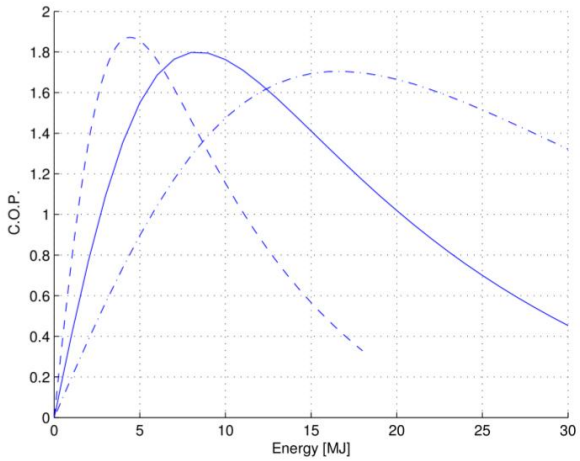

Figure 4: Examples of COP curves of different chillers. The COP is reported as a function of the cooling energy request.

size	a_1	a_2	a_3	a_4	c_1	c_2	c_3
small	5.6^{-4}	10.11	7.00	0.9327	2.49^{-4}	4.98^{-2}	1.26
medium	1.09^{-3}	20.22	3.80	0.9327	3.79^{-5}	2.77^{-2}	2.46
large	2.3^{-3}	40.44	1.98	0.9327	3.56^{-6}	1.58^{-2}	5.11

Table 1: Possible values of the chiller parameters.

Chiller convex biquadratic approximation. A convex approximation of the non linear Ng-Gordon model can be derived from the biquadratic expression

$$E_{ch,l} = c_1 E_{ch,c}^4 + c_2 E_{ch,c}^2 + c_3. \tag{31}$$

Weighted least square can be used in order to best fit the most relevant points, i.e, those that correspond to 0 energy request and to the maximum COP value.

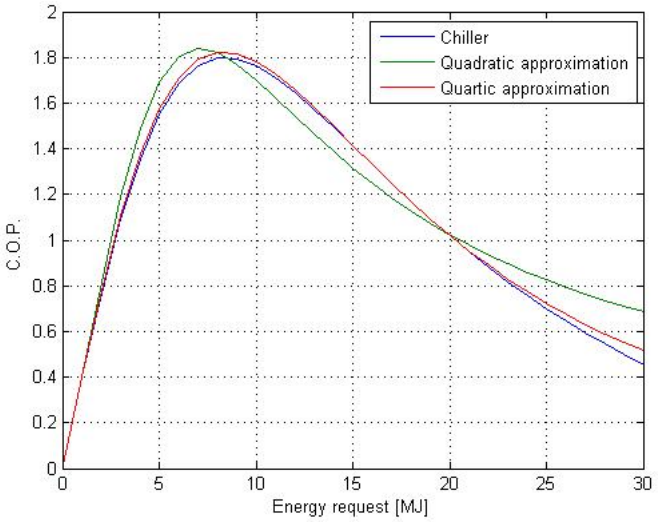

Figure 5: Chiller COP curve: comparison between the quadratic and the biquadratic approximations.

Figure 5 shows that the biquadratic approximation fits very well the real function all over the range, and especially at the best COP points. The error percentage is depicted in Figure 6. The second order polynomial approximation, despite its simplicity, lacks in precision so that it should not be used for modeling.

Chiller convex piecewise affine approximation. Another possible convex approximation of the nonlinear Gordon-Ng equation is via a PieceWise Affine (PWA) function. In this case,

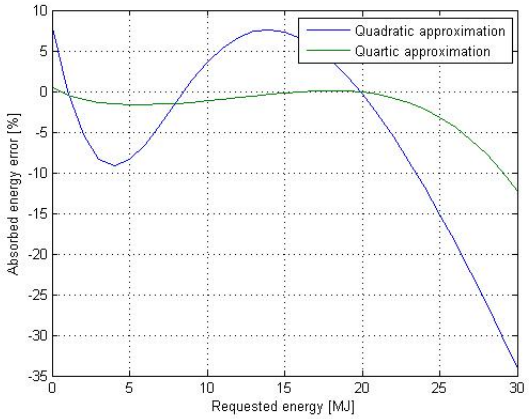

Figure 6: Approximation error in terms of absorbed electrical energy as a function of the cooling energy request.

the nonlinear characteristic is approximated by a finite number of affine terms and their convex envelope that is generated applying the max operator to the set of affine terms. If we collect the coefficients of the affine terms in vectors m_c and q_c, the piecewise approximation can be compactly written as:

$$E_{ch,\ell}(k) = \max\{m_c E_{ch,c}(k) + q_c\},$$

where the max operator should be applied componentwise.

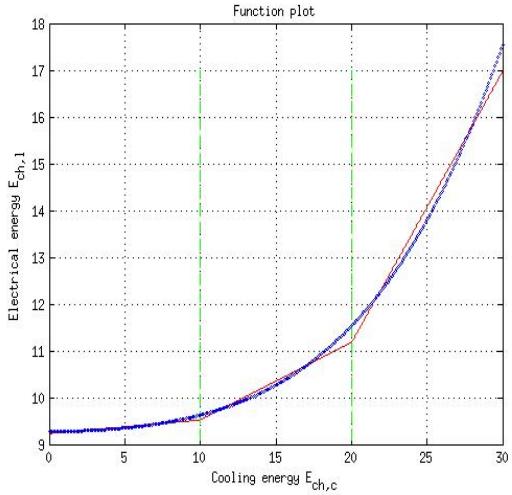

Figure 7: PWA approximation: electrical energy consumption as a function of the cooling energy request.

Note that a more precise model can be obtained by using a finer piecewise affine approxi-

mation (Figure 7).

Since the cooling energy request from a building is affine as a function of the zones temperatures, then, the joint model of the building and chiller becomes a PWA function of the zones temperatures. In turn, logical conditions making use of binary variables can be introduced to describe the switching between the different affine terms in the PWA function, thus leading to a mixed integer affine description.

On-off switching. In order to have the possibility of switching on and off the chiller, we introduce the binary variable $\delta_{ch}(k)$, $k = 1, \dots, M$, that represents the on ($\delta_{ch}(k) = 1$) and off ($\delta_{ch}(k) = 0$) status of the chiller at time k, $k = 1, \dots, M$. The control inputs to the chiller are then the cooling energy request $E_{ch,c}(k)$ and the on/off command $\delta_{ch}(k)$, for each time slot $k = 1, \dots, M$, which are related through the logical condition

$$\delta_{ch}(k) = 1 \Leftrightarrow E_{ch,c}(k) > 0. \tag{32}$$

Interestingly, (32) can be expressed as a mixed integer linear condition using the *Conjunctive Normal Form* in [6]:

$$E_{ch,c}(k) \geq \varepsilon - \varepsilon \cdot (1 - \delta_{ch}(k)) \tag{33}$$

$$E_{ch,c}(k) \leq E \cdot \delta_{ch}(k), \tag{34}$$

where E is an upper bound on u_{ch} and ε is a small quantity, typically chosen equal to the machine precision. Equations (33) and (34) lead to the relations

$$\delta_{ch}(k) = 0 \Leftrightarrow E_{ch,c}(k) = 0$$
$$\delta_{ch}(k) = 1 \Leftrightarrow E_{ch,c}(k) \in [\varepsilon, E),$$

that are practically equivalent to (32).

We can rewrite the model of the chiller integrating the on-off condition as

$$E_{ch,\ell}(k) = (c_1 E_{ch,c}^4 + c_2 E_{ch,c}^2 + c_3)\delta_{ch}(k),$$

for the biquadratic formulation, and

$$E_{ch,\ell}(k) = \max\{m_c E_{ch,c}(k) + q_c\}\delta_{ch}(k),$$

for the PWA formulation. The latter is particularly convenient since a product between a (piecewise) linear model $Mx + q$ and a discrete variable δ can be easily reduced to a mixed integer linear condition via the big M procedure, i.e., by introducing an auxiliary variable

$$z = \delta \cdot (M \cdot x + Q)$$

subject to

$$\begin{aligned} z &\leq M \cdot x + Q + (1 - \delta) \cdot BigM \\ z &\leq \delta \cdot BigM \\ z &\geq 0 \end{aligned}$$

where $BigM$ is an upper bound on $Mx + q$.

3.1.3 Combined heat and power unit

A Combined Heat and Power (CHP) unit is a device that jointly produces electricity and heat power while consuming primal energy (i.e. fossil fuels or hydrogen) with the purpose of reducing the amount of energy wasted in the ambient. In most cases one of these two products is a byproduct. For example, modern power plants recover scattered heat and deliver it for district heating purposes. Big sized CHPs are becoming widely used and highly performing. At the same time a large number of micro-CHP solutions are being developed, the most promising ones being the microturbines, that convert gas into heat and electricity. Combined Cooling, Heat and Power (CCHP) devices are also available that convert part of the produced heat into cooling energy.

CHP devices are currently being used for peak shaving during peak load periods. Jointly with thermal and/or electrical storing systems, they can be thought of as an effective means to actively regulate the electrical status of a smart grid, balancing power demand and generation.

We consider here a microturbine and model it through two static characteristics describing the electrical power production and the heat production, both as a function of the fuel volumetric flow rate. Possible curves are plotted in Figure 8 that refers to the C30 microturbine produced by Capstone company [11], whose data are reported in Table 2. We can see that both curves are almost linear. The electrical energy $E_{mt,\ell}(k)$ and the heat $E_{mt,h}(k)$ produced by this microturbine during the time slot k can then be expressed as affine functions of the

fuel volumetric flow rate $u_{mt}(k)$, that is supposed to be constant in each time slot

$$E_{mt,\ell}(k) = m_\ell u_{mt}(k) + q_\ell$$
$$E_{mt,h}(k) = m_h u_{mt}(k) + q_h,$$

where m_ℓ, q_ℓ, m_h, and q_h are positive coefficients suitably defined so as to match data in Table 2)

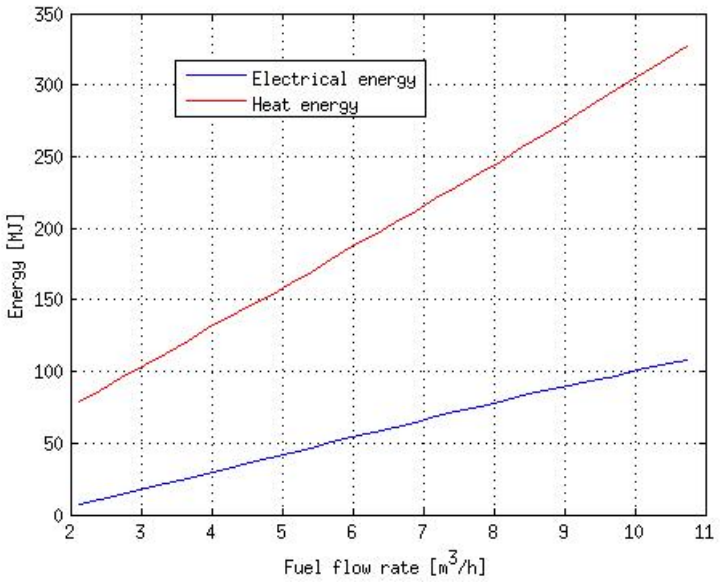

Figure 8: CHP Partial load performance

The microturbine specifications require a minimum fuel volumetric flow rate $u_{mt}^{\min}$ for the unit to be operative. We can introduce the possibility to switch the unit on or off, which is modeled by the binary variable δ_{mt} with the following logical condition associated:

$$u_{mt}(k) \leq u_{mt}^{\min} \Leftrightarrow \delta_{mt}(k) = 0.$$

Such condition can be practically implemented by the means of the relations:

$$u_{mt}(k) \leq \delta_{mt}(k) u_{mt}^{\max} + (1 - \delta_{mt}(k)) u_{mt}^{\min}$$

$$u_{mt}(k) \geq \delta_{mt}(k)(u_{mt}^{\min} + \varepsilon)$$

where $u_{mt}^{\max}$ ie the maximum flow rate and $\varepsilon > 0$ is set equal to the machine precision.

The microturbine model thus becomes

$$E_{mt,\ell}(k) = \delta_{mt}(k)(m_\ell u_{mt}(k) + q_\ell)$$
$$E_{mt,h}(k) = \delta_{mt}(k)(m_h u_{mt}(k) + q_h).$$

Note that here we do not model explicitly the microturbine transient from on to off, as it instead suggested in [15], but we assume that its duration is smaller than the time slot length Δt. For control purposes, we have two inputs that can possibly be set: the fuel volumetric flow rate $u_{mt}(k)$ and the on/off status of the microturbine $\delta_{mt}(k)$, $k = 1, \ldots, M$.

Fuel consumption $[\frac{m^3}{s}]$	Nominal power [kW]	Electrical energy $[MJ_E]$	Thermal energy $[MJ_T]$
2.123	2	7.20	78.81
2.425	3	10.80	86.40
2.725	4	14.40	95.16
3.052	5	18.00	104.13
3.352	6	21.60	112.89
3.679	7	25.20	121.33
3.952	8	28.80	129.77
4.224	9	32.40	137.15
4.524	10	36.00	145.59
4.824	11	39.60	152.98
5.124	12	43.20	161.42
5.424	13	46.80	169.86
5.696	14	50.40	178.30
5.996	15	54.00	186.74
6.296	16	57.60	195.18
6.596	17	61.20	203.62
6.896	18	64.80	212.06
7.168	19	68.40	220.50
7.468	20	72.00	228.94
7.795	21	75.60	238.44
8.095	22	79.20	246.88
8.395	23	82.80	256.37
8.722	24	86.40	265.87
9.049	25	90.00	275.36
9.376	26	93.60	285.92
9.730	27	97.20	296.47
10.030	28	100.80	305.96
10.385	29	104.40	315.46
10.739	30	108.00	327.06

Table 2: Microturbine Capstone C30, nominal values.

3.1.4 Storage

Thermal Energy Storages (TESs) are becoming widely used in medium size grids. TESs represent the most effective way, or even sometimes the only way, to take advantage of renewable energy sources. This is indeed the case for thermal solar energy and geothermal energy systems. In a smart grid context, they can be used as energy buffers for unbinding energy production from energy consumption. More specifically, in a building cooling scenario, a TES for cooling energy can

- shift the production of cooling energy to off-peak hours of electrical energy consumption;
- allow chillers to operate in high-efficiency conditions;
- smooth peaks of electrical energy request with benefits both for power production and distribution network systems.

A thermal storage is operated according to two main strategies:

- if it is big enough to supply all the energy needed for cooling the building, e.g., during office hours when the building is actually occupied, one can define a working period in which the storage provides all the requested cooling energy, and a off period in which the storage is fully charged by operating the chiller at its maximum efficiency, typically at night. The main advantages of this strategy are that electrical energy consumption is avoided during the day and the control logic is simple to be implemented. The chiller and storage sizes can be set based on an average load request, although it is often over-dimensioned as a strategy to provide robustness to disturbances;
- the storage is operated jointly with the chiller, so as to make the chiller work in the most efficient conditions and eventually be switched off if this reduces the costs. In this case, it is more difficult to define the charge/discharge policy and to set the appropriate size of both the chiller and the storage.

There are many different technical solutions to store thermal energy, the most widely used are indeed fluid tanks and phase changing materials based storages. We shall next briefly present models for a fluid tank:

- *fully mixed*, where the temperature in the storage is supposed to be homogenous and one single differential equation can be used, i.e.,

$$C_s \frac{dT_s}{dt} = \dot{m} c_p (T_{inlet} - T_s) - K_{out}(T_s - T_{outlet}),$$

  where C_s denotes the thermal capacity of the storage, T_s the temperature of the fluid, $\dot{m}$ the mass flow and K_{out} the heat exchange coefficient of the external surface with the environment. As for T_{inlet} (T_{outlet}), it represents the inlet (outlet) water temperature;

- *fully stratified*, where the storage is modeled as divided in a certain number of different and non-mixing layers, each one with a different temperature and different heat exchange rate with the external environment but none with adjacent layers of fluid. In this case, when an inlet water temperature variation happens, then the inlet water occupies the position where its temperature is the closest to adjacent layers;

- *stratified*, proposed by *Sharp*[37], where, unlike the fully stratified one, the heat exchanges between adjacent layers are taken into account. Even if the turbulence effect is not explicitly considered, by increasing the number of layers it is possible to obtain temperature profiles that are similar to those obtained when turbulence is accounted for;

- *Gahajar*'s model [43], where a constant inlet water temperature is assumed but the effects of turbulence is modeled. This is done by varying the diffusivity coefficient depending on the storage geometric and thermodynamic characteristics like the fluid flow, Reynolds and Richardson numbers and the shape and position of the inlet. Gahajar's model is one that better predicts the shape of the thermocline.

- *Black Box* model, derived based on system identification techniques, with the energy exchange (drowned or inserted) as input and the thermal energy stored as output. This way of modeling does not consider the way energy is stored or provided.

The simplest model is a first order AutoRegressive eXogenous (ARX) system:

$$S(k+1) = aS(k) - s(k)$$

where $S(k)$ is the amount of cooling energy stored and $s(k)$ is the cooling energy exchanged ($s(k) > 0$ if the storage is discharged, and $s(k) < 0$ if it is charged), in time slot k, while $a \in (0, 1)$ is a coefficient introduced to model energy losses.

Based on this model we can reformulate the thermal storage dynamics in a compact form as

$$\boldsymbol{S} = \Xi_0 S(0) + \Xi_1 \boldsymbol{s},$$

where we set $\boldsymbol{S} = [S(1) \cdots S(M)]^\top$, $\boldsymbol{s} = [s(0) \cdots s(M-1)]^\top$, and Ξ_0 and Ξ_1 are suitable matrices. In this case $\boldsymbol{s}$ represents a control input. If $s(k) > 0$, the storage supplies part of the cooling energy at k, thus reducing the amount of cooling energy requested to the chiller, whereas if $s(k) < 0$, then the chiller has to produce additional cooling energy that is stored.

Remark 1 (active and passive storages) *The described systems are active thermal storages, that are directly operated by charge/discharge commands. Passive thermal storages are instead physical elements, like the walls of a building, that can accumulate and release thermal energy but are not directly charged/ discharged.*

Remark 2 (batteries) *Batteries accumulating/releasing electrical energy can still be modeled via an ARX first order system.*

3.1.5 Renewable energy generator

Wind turbine. A wind turbine is used to convert the kinetic energy of air mass in motion to electric energy. For the application of a wind turbine as a renewable energy generator, a detailed hybrid model can be found in Section 2 of this document.

3.2 Configurations of the smart grid

In this section, we show how to compose the models introduced in Section 3.1 in order to define a (scalable) smart grid. Model composition is indeed easy given that the different components are described through the energy (thermal or electrical) that they need as input or provide as output. Energy balance equations can then be adopted where signals can be simply summed up (e.g., the cooling requests of different buildings) and/or used to feed other models (e.g., the chiller receive as input the cooling energy request of multiple buildings and provide as output the corresponding electrical energy consumption).

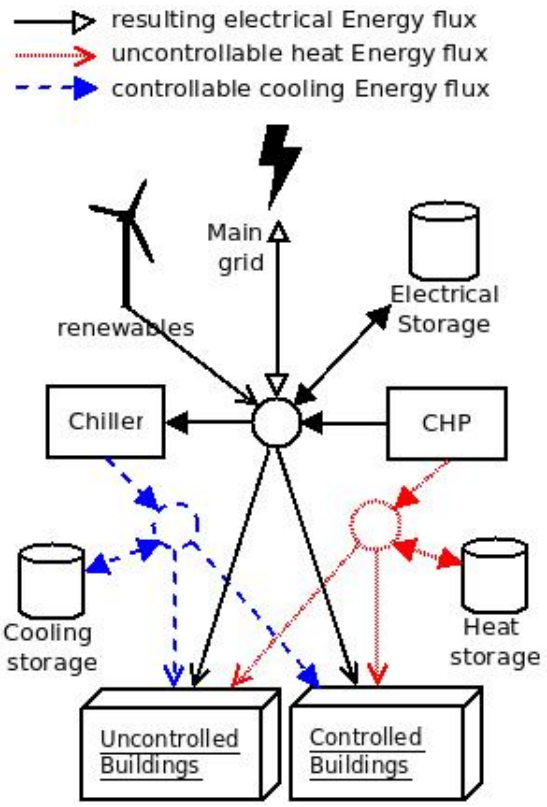

Figure 9: Overall Grid Structure: The micro-grid is composed by different buildings sharing different resources. Fluxes related to the three different kinds of energies (electrical, heat, and cooling energies) are associated with a different style of the line: black thick for electrical energy; red thin for heat energy; blue dotted for cooling energy. The three nodes appearing in the figure do not correspond to any physical component but are introduced to point out that fluxes associated with the same kind of energy add up to zero. Different arrowheads are used depending on the fact that the corresponding energy flux can be controlled (e.g., that related to a storage unit), controlled only indirectly (e.g., the electrical energy requested by the chiller), not controlled (e.g., renewable energy production).

Figure 9 shows a possible micro-grid configuration and the energy fluxes between the grid components. The micro-grid is composed by a number of different buildings that share common resources such as cooling and heat storages, chillers, CHP units, batteries and renewable energy generators. Electrical, heat, and cooling energies sum up in three nodes and the nodes balance is zero. Some energy contributions can be controlled (e.g., those related to storage units), some others not (renewable energy production). This is pointed out using different arrowheads in Figure 9. As for buildings, some of them are named as controlled in the sense that, through a suitable choice of the zones temperatures, they can modulate their cooling

energy demand and contribute to the micro-grid optimization. If the zone temperatures are fixed and given by some comfort profiles, then the building is uncontrollable.

The electrical energy exchanged with the main grid is given by the amount of energy needed to maintain the balance between electrical energy demand and generation. A micro-grid becomes "smart" when the possibility to control the grid status via a suitable choice of the controllable variables according to some criterion is introduced. The most natural criterion is to achieve the minimum cost while guaranteeing the satisfaction of users energy needs. Costs are mainly due to the electrical energy exchange[1] with the main grid and by other operation costs such as devices startup and fuel consumption. The overall cost over the time horizon is thus:

$$J = \phi E_L + C_{mt} + C_{ch} + C_f \tag{35}$$

where

- $\phi = [\phi(1), \cdots, \phi(M)]$, $\phi(k)$ being the electricity cost in time slot k;
- $E_L = [E_L(1), \cdots, E_L(M)]^T$, $E_L(k)$ being the electrical energy exchange with the main grid in the time slot k;
- $C_{mt} = \sum_{k=1}^{M} C_{mt}(k)$ cost for microturbine (CHP) startup;
- $C_{ch} = \sum_{k=1}^{M} C_{ch}(k)$ cost for chiller startup;
- $C_f = \sum_{k=1}^{M} C_f(k)$ cost of microturbine fuel consumption.

Note that the cost formulation may involve introducing logical constraints. For example, the startup cost introduced to avoid continuous (and unrealistic) switchings of the chiller can be modeled as follows:

$$C_{ch}(k) = C_{ch}^{\text{on}} \cdot \max\{\delta_{ch}(k) - \delta_{ch}(k-1), 0\}$$

where C_{ch}^{on} is the actual startup cost which is accounted for at time k only if the chiller was off at $k-1$ and is switched on at k, that is $\delta_{ch}(k) - \delta_{ch}(k-1) = 1$. Similarly, for the microturbine we have

$$C_{mt}(k) = C_{mt}^{\text{on}} \cdot \max\{\delta_{mt}(k) - \delta_{mt}(k-1), 0\}. \tag{36}$$

[1] We model revenues as negative costs

The fuel costs of the microturbine are proportional to the amount of fuel consumption during the k-th time slot, i.e.,

$$C_f(k) = \psi_f \cdot \delta_{mt}(k) \cdot u_{mt}(k) \cdot \Delta t,$$

where ψ_f is the unitary cost of the fuel.

To describe E_L for every possible configuration let shortly define components trough letters as: Building (B), Chiller (C), Storage (S) Microturbine (M) and the overall electrical load (E_L). We define then as superscripts the modeling type according to Tables 3 and 4, and as subscripts the considered output if there are many (i.e. M_h^B stays for heat production microturbine linear on-off model).

According to this notation we can derive the expression for the electrical energy exchange with the main grid as:

$$E_L = C\Big(B^B + B^A + S_c\Big)^{[?]} + M_l^{[?]} + S_e \tag{37}$$

where [?] stays for model type choice, e refer to electrical quantities (i.e. S_e is a battery), c stays for cooling and h for heating. Notation $C(\cdot)$ is used for the chiller that receives as input the sum of the cooling energy requests by the buildings and thermal storage.

If we plug (37) into equation (35), we finally get:

$$J = \phi \left[C\Big(B^B + B^A + S_c\Big)^{[?]} + M_l^{[?]} + S_l \right] + C_{mt} + C_{ch} + C_f.$$

When we compose a grid model plugging together all the elements, we also get a number of constraints associated with them. Constraints express both technical limits (i.e. maximum cooling energy that a chiller can provide) and performance requirements (i.e. comfort temperature range). Additional constraints can be added if needed (e.g. the maximum amount of electrical energy that the main grid can provide).

Since various disturbance signals are present and have been accounted for in modeling the smart grid components, different control design strategies (certainty equivalence based, robust, stochastic) can be adopted for (average) cost minimization in presence of (nominal, robust, probabilistic) constraints.

Example 1. In [22], the model configuration in Figure 10 is adopted, with

$$E_L = C\Big(B^A + S_c\Big)^A.$$

Nominal disturbances are considered in certainty equivalence based control design, which reduces to solving a constrained convex optimization program. The purpose of [22] is showing the role of the building structure as a passive thermal storage.

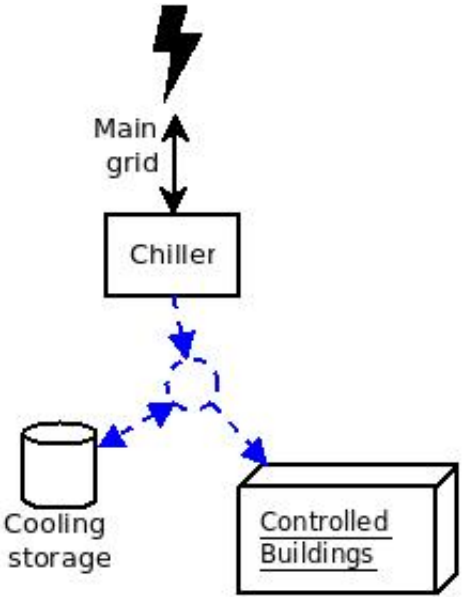

Figure 10: Example 1: Configuration

Example 2. In [21], the model configuration in Figure 11 is adopted, with

$$E_L = C\Big(B^B + S_c\Big)^D + M_l^{[B]} + S_l.$$

The on-off switching of the chiller is introduced to add more flexibility. Nominal disturbances are considered in certainty equivalence based control design, which involves solving a mixed integer linear programming problem due to the presence of the on-off switching command.

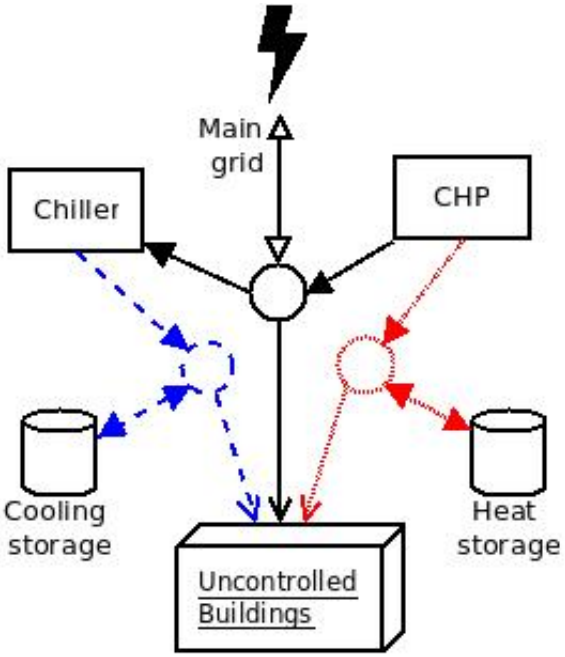

Figure 11: Example 2: Configuration

Chiller

Type	Description	Model	Op.prob
A	biquadratic approx.	$E_{ch,l} = c_1 E_{ch,c}^4 + c_2 E_{ch,c}^2 + c_3$	convex-NL
B	piecewise approx.	$\max\{m_c E_{ch,c} + q_c\}$	MILP
C	biquadratic on-off	$(E_{ch,l} = c_1 E_{ch,c}^4 + c_2 E_{ch,c}^2 + c_3)\delta_{ch}$	MINLP
D	piecewise on-off	$(\max\{m_c E_{ch,c} + q_c\})\delta_{ch}$	MINLP*
Var	**Type**	**Description**	**Domain**
$E_{ch,l}$	output	Absorbed electrical energy	Real
$E_{ch,c}$	control input	Cooling energy request	Real
δ_{ch}	control input	On-off logical status	Binary
Type	**Constraint**	**Equations**	**Var**
A-D	Electricity bounds	$0 \leq E_{ch,l} \leq E_{ch,l}^{\max}$	output
A-D	Cooling energy bounds	$0 \leq E_{ch,c} \leq E_{ch,c}^{\max}$	input
C-D	Logical on-off	$E_{ch}(k) \geq \varepsilon - \varepsilon \cdot (1 - \delta_{ch}(k))$ $E_{ch}(k) \leq E \cdot \delta_{ch}(k))$	input

Building

Type	Description	Model	Op.prob
A	Linear Controllable	$E_{r,c} = BT_z + Wd + b$	LP
B	Linear Simulator	$E_{r,c} = B\bar{T_z} + Wd + b$	/
Var	**Type**	**Description**	**Domain**
T_z	control input	Zone temperatures	Real
d	uncontrollable input	Disturbances	Real
$E_{r,c}$	output	Cooling energy request by building	Real
$E_{c,j}$	output	Cooling energy request by zone j	Real
Type	**Constraint**	**Equations**	**Var**
A	Comfort	$T_z^{min} \leq T_z \leq T_z^{\max}$	input
A	Cooling energy bounds	$0 \leq E_{r,c} \leq E_{r,c}^{\max}$ $0 \leq E_{c,j} \leq E_{c,j}^{\max}$	output

Table 3: Proposed models 1/2

MT

Type	**Description**	**Model**	**Op.prob**
A	linear	$E_{mt,\ell}(k) = m_\ell u_{mt}(k) + q_\ell$ $E_{mt,h}(k) = m_h u_{mt}(k) + q_h$	LP
B	linear on-off	$E_{mt,\ell}(k) = \delta_{mt}(k)(m_\ell u_{mt}(k) + q_\ell)$ $E_{mt,h}(k) = \delta_{mt}(k)(m_h u_{mt}(k) + q_h)$	MINLP*
Var	**Type**	**Description**	**Domain**
$E_{mt,l}$	output	Produced electricity	Real
$E_{mt,h}$	output	Produced heat	Real
u_{mt}	control input	fuel inlet	Real
δ_{mt}	control input	On-off logical status	Binary
Type	**Constraint**	**Equations**	**Var**
A-B	Fuel inlet bounds	$0 \leq u_{mt} \leq u_{mt}^{\max}$	output
A-B	Cooling energy bounds	$0 \leq E_{ch,c} \leq E_{ch,c}^{\max}$	input
C-D	Logical on-off	$u_{mt}(k) \leq \delta_{mt}(k) u_{mt}^{\max}(k) + u_{mt}^{\min}(k)$ $u_{mt}(k) \geq \delta_{mt}(k)(u_{mt}^{\min}(k) + \varepsilon)$	input

Storage

Type	**Description**	**Model**	**Op.prob**
A	Storage	$\boldsymbol{S} = \Xi_0 S(0) + \Xi_1 \boldsymbol{s}$	/
Var	**Type**	**Description**	**Domain**
$\boldsymbol{S}$	State	Energy content	Real
$\boldsymbol{s}$	control input	Energy exchange	Real
Type	**Constraint**	**Equations**	**Var**
A	Energy content bounds	$0 \leq \boldsymbol{S} \leq \boldsymbol{S}^{\max}$	state
A	Energy exchange bounds	$\boldsymbol{s}^{min} \leq \boldsymbol{s} \leq \boldsymbol{s}^{\max}$	state

Table 4: Proposed models 2/2

4 Modelling of automated vehicle safety aspects

UnCoVerCPS presents the automated driving application as a cyber-physical system composed of two automated vehicles, both moving in a driving scenario on a collaborative basis. The objective of the on-the-fly verification in terms of safety of this application is to guarantee that the vehicles are able to conduct manoeuvres correctly and in a safe manner, while interacting within a complex 'take-over scenario with an obstacle in one of the lanes'. Both vehicles, an automated Renault Twizy and an automated Volkswagen Passat *'FasCar'* have been provided to the project with already autonomous driving capabilities. Next sections present the models of the automated vehicles that describe the dynamics and trajectory followers. For the implementation of the collaborative scenario, a new *vehicle to vehicle communication model* has been developed within UnCoVerCPS. The communication model is presented in section 4.8.

4.1 Vehicle model for lateral dynamics

The model used in both vehicles of UnCoVerCPS has been designed based on a bicycle model, as presented by [33], with three degrees of freedom. The dynamics are represented by the vehicle lateral position y, measured along the lateral axis of the vehicle, and the vehicle yaw angle ψ, measured with respect to the global X axis, as depicted in figure 12:

The vehicle lateral dynamics are obtained applying Newton's second law for the motion along the lateral axis, and a moment balance around the z axis for the yaw dynamics of the vehicle as shown in (38).

$$\begin{aligned} ma_y &= m(\ddot{y} + V_x\dot{\psi}) = F_{yf} + F_{yr} \\ I_z\ddot{\psi} &= aF_{yf} - bF_{yr} \end{aligned} \tag{38}$$

The lateral forces F_{yf} and F_{yr} have to be chosen to solve (38). The results presented in [33] show that the lateral tire force of a tire is proportional to the slip-angle for small slip-angles. Thus, lateral forces are defined as shown in Equations (39)-(40).

$$F_{yf} = 2C_{\alpha f}\alpha f \tag{39}$$

$$F_{yr} = 2C_{\alpha r}\alpha r \tag{40}$$

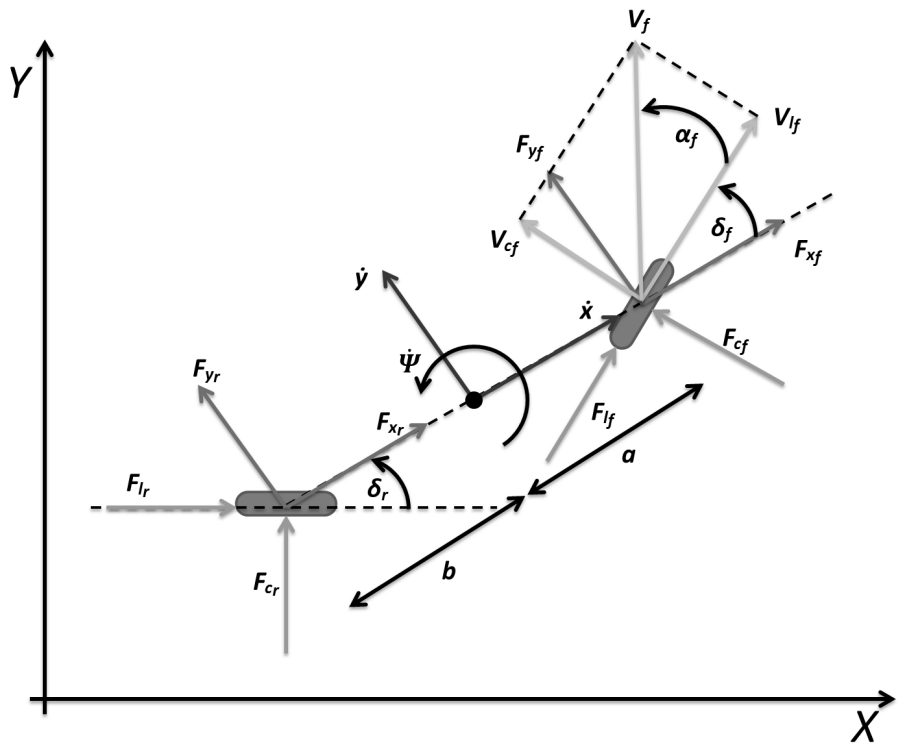

Figure 12: Bicycle model of the Twizy vehicle

where $C_{\alpha f}$ and $C_{\alpha r}$ represent the front and rear wheel cornering stiffness respectively and αf and αr are the front and rear slip angle.

The equations of the lateral dynamic of the bicycle model as shown in (41) and (42) are obtained by combining Equations (38) to (40):

$$\ddot{y} = -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}\dot{y} + \left(-\frac{2\alpha C_{\alpha f} - 2bC_{\alpha r}}{mV_x} - V_x \right)\dot{\psi} + \frac{2\alpha C_{\alpha f}}{m}\delta \tag{41}$$

$$\ddot{\psi} = -\frac{2\alpha C_{\alpha f} + 2C_{\alpha r}}{I_z V_x}\dot{y} + \left(-\frac{2a^2 C_{\alpha f} - 2b^2 C_{\alpha r}}{I_z V_x} \right) + \frac{2\alpha C_{\alpha r}}{I_z}\delta, \tag{42}$$

where m is the vehicle mass, V_x is the longitudinal speed, I_z is the yaw inertia, a and b are the distance between the front/rear wheels and the center of gravity, respectively, and δ represents the front wheel steering angle (which is the control signal in the autonomous control approach). An additional, important factor regarding the slope of the road might be added, as presented in [10], resulting in an updated equation related to the vehicle's lateral acceleration as defined

in (43). This factor is to be taken into account in terms of safety, considering the maximum speed of the vehicle in curves.

$$\ddot{y} = -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}\dot{y} + \left(-\frac{2\alpha C_{\alpha f} - 2bC_{\alpha r}}{mV_x} - V_x \right)\dot{\psi} + \frac{2\alpha C_{\alpha f}}{m}\delta - \frac{ge}{100} \tag{43}$$

Combining (41) and (42), a state space model can be specified as described in (44). This way, the vehicle dynamic equations can be used to estimate predictions on future positions of the vehicle.

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) \end{aligned} \tag{44}$$

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & V_x \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & -\frac{2\alpha C_{\alpha f}-2bC_{\alpha r}}{mV_x} - V_x & 0 \\ 0 & -\frac{2\alpha C_{\alpha f}+2C_{\alpha r}}{I_z V_x} & -\frac{2a^2 C_{\alpha f}-2b^2 C_{\alpha r}}{I_z V_x} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2\alpha C_{\alpha f}}{m} \\ \frac{2\alpha C_{\alpha r}}{I_z} \\ 0 \end{bmatrix}\delta \tag{45}$$

4.2 Variations of the vehicle model

The following bicycle model adds considerations about the tire load, which allows to remove (variable) mass terms from the equations, when one assumes the maximum transmissible tire force to be proportional to the tire load in a limited region around the tire's nominal load. In order to maintain the number of relevant state dimensions, the suspension is assumed to be infinitely stiff. The load / normal forces on the front and rear tires $F_{z,f}, F_{z,r}$ are then directly influenced by the overall braking/acceleration force F_x:

$$F_{zf} = \frac{b}{L}mg - \frac{h}{L}F_x, \qquad F_{zr} = \frac{a}{L}mg + \frac{h}{L}F_x, \tag{46}$$

with $L = a+b$ the distance between the front and rear axle. Using the proportional relationship between lateral force and tire load $F_{yf} \approx C_f \mu \frac{F_{zf}}{F_{z0}} \cdot \alpha_f$, a parameter change to the relative stiffness $c_f = C_f / F_{z,0}$, the lateral tire forces can be formulated as:

$$F_{yf} = \mu c_f \left(\frac{b}{L}mg - \frac{h}{L}F_b \right)\alpha_f, \qquad F_{yr} = \mu c_r \left(\frac{a}{L}mg + \frac{h}{L}F_b \right)\alpha_r \tag{47}$$

The equations of motion for the center of gravity (COG) are then:

$$f_1 = \dot{v}_x = a_x + v_y\omega \tag{48}$$

$$f_2 = \dot{v}_y = \mu\left(\left(\frac{b}{L}g - \frac{h}{L}a_x\right)c_f\alpha_f + \left(\frac{a}{L}g + \frac{h}{L}a_x\right)c_r\alpha_r\right) - v_x\omega \tag{49}$$

$$f_3 = \dot{\omega} = \frac{m}{I_z}\mu\left(a\left(\frac{b}{L}g - \frac{h}{L}a_x\right)c_f\alpha_f - b\left(\frac{a}{L}g + \frac{h}{L}a_x\right)c_r\alpha_r\right) \tag{50}$$

This model contains the invariant and well known[2] parameters L (wheelbase) and g (gravitational constant), as well as the five variant paramters $\theta = [\mu, \frac{I_z}{m}, \frac{b}{L}, c_f, c_r]$: Friction coefficient, ratio of mass to rotational inertia, relative position of center of gravity, relative front tire stiffness as well as relative rear tire stiffness.

4.3 Estimation of model parameters using a static approach

In the following a first estimate of the model parameters is computed by comparing measured accelerations to model-predicted accelerations for each sample of a recorded time series. In order to find the best parameter fit θ^{LS}, the error $e^i(x^i, \theta)$ is defined for a point of time t^i and a measured sample $x^i = [v_x^i, v_y^i, \omega^i, \dot{\omega}^i, a_x^i, a_y^i, \delta^i]$:

$$e(x^i, \theta) := \begin{pmatrix} a_y^i - \left(f_2(v_x^i, v_y^i, \omega^i, a_x^i, \delta^i;\ \ \theta) + v_x\omega\right) \\ \dot{\omega}^i - f_3(v_x^i, v_y^i, \omega^i, a_x^i, \delta^i;\ \ \theta) \end{pmatrix} \tag{51}$$

The most fitting parameter vector θ^{LS} is computed as the value, which minimizes the sum of squared errors over all points of time:

$$\theta^{\mathrm{LS}} = \underset{\theta}{\operatorname{argmin}} \sum_{1<i<N} \|e(x^i, \theta)\|_2^2 \tag{52}$$

To record the measurements, the vehicle is manually driven and the driver's inputs, steering angle δ and longitudinal acceleration a_x are recorded, as well as observable variables $[X, Y, \psi, \dot{X}, \dot{Y}, \omega, a_y]$. For the measurements a Novatel SPAN-CPT inertial navigation system with dual GPS antennas, (using the additional Flex6 GPS receiver), and differential correction of GPS signals is used. The secondary GPS receiver and antenna measure the vehicle's heading with Novatel's ALIGN feature, which allows us to compute a slip angle estimate. Control

[2] g is earth gravity and does not require measurement. L is a well known parameter in the sense that it can be easily measured and is time-invariant. Whereas other parameters such as rotational inertia I_z or tire stiffness $c_{f|r}$ are cumbersome to measure directly and are time-variant.

inputs are recorded from the vehicle CAN bus. All variables are recorded at 100Hz. The slip angle estimate and the rotational acceleration are computed as:

$$v_x^i = \cos(\psi^i)\dot{X}^i + \sin(\psi^i)\dot{Y}^i \tag{53}$$

$$v_y^i = -\sin(\psi^i)\dot{X}^i + \cos(\psi^i)\dot{Y}^i \tag{54}$$

$$\dot{\omega}^i \approx \frac{\Delta}{\Delta t}\omega^i \tag{55}$$

For the test drive we selected a course with bends for constant cornering and a slalom course for more dynamic excitation of the vehicle, see Fig. 13.

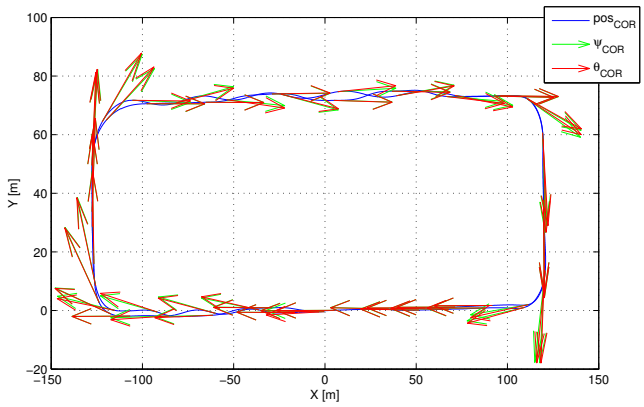

Figure 13: Manually driven test track for parameter identification.

For the analysis, the recording was restricted to those points of time, where the velocity was above 10m/s: The time series is depicted in Figure 14.

$\frac{I_z}{m}$	$\frac{b}{L}$	c_f	c_r	(h)	(L)	(g)	(μ)
1.57	0.57	-10.8	-17.8	0.5	2.7	9.81	1

Table 5: Estimated vehicle parameters θ^{LS}. (The parameters in brackets are not evaluated and are taken as given.)

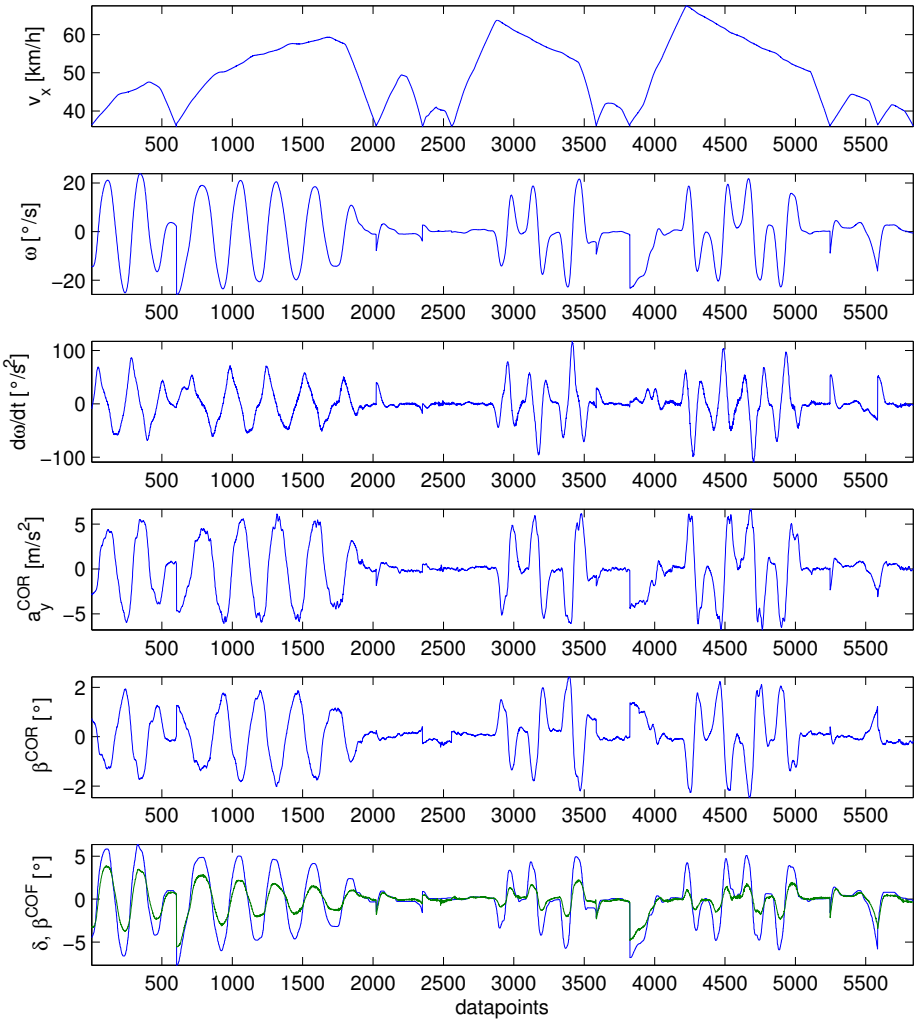

Figure 14: Recorded time series for vehicle state.

The parameters resulting from the fitting method are given in Tab. 5.

A comparison between observations and model predictions with the parameters θ^{LS} are listed in Fig. 15 for the tire forces. The tire forces are computed based on the observations a_x, a_y, $\dot{\omega}$ and the model parameters μ, h, m/I_z and b/L. The figure shows a ratio between tire lateral force and normal force, which corresponds to the force utilization in the interval $[-\mu, \mu] = [-1, 1]$. Each dot is a pairing of slip angle and resulting force utilization. The slopes of the fitted curves correspond to the parameters c_f and c_r.

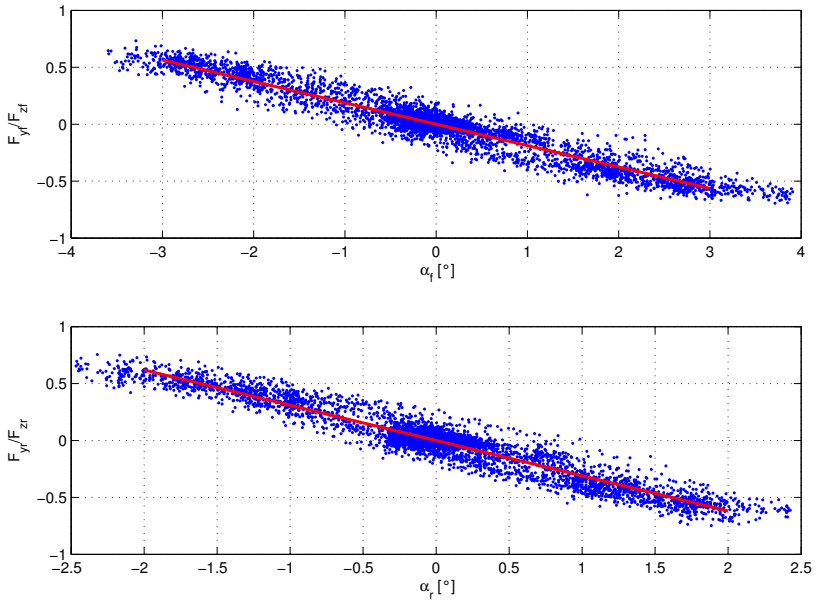

Figure 15: Comparison of tire forces estimated from recorded accelerations (blue) and linear tire forces predicted by vehicle model with θ^{LS} (red).

Fig. 16 provides a comparison between measured accelerations and predicted accelerations. Predicted accelerations are computed based on the model function f with the measured state $[v_x, v_y, \omega]^T$ and the input δ.

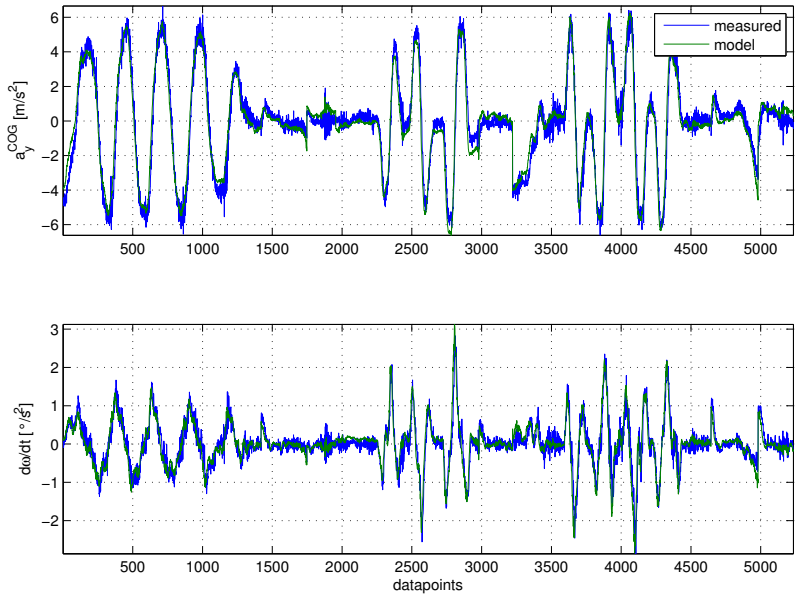

Figure 16: Comparison of recorded accelerations (blue) and accelerations predicted by model with parameter θ^{LS} (green).

The results look promising, yet the influence of measurement noise and process noise cannot be distinguished by the static identification method. Therefore, a more precise evaluation of the conformance between model and real system is required.

4.4 Parameter estimation using a dynamic Likelihood Maximization approach

The concept of Maximum Likelihood (ML) estimation describes a way to handle parameter identification for nonlinear systems with hidden (not directly measurable) states. In principle, the distributions of sensor noise and state disturbances can be explicitly taken into account. In practice, the full ML problem cannot be solved in acceptable computation time, therefore approximation schemes are employed. The student thesis [38] has been supervised in the course of UnCoVerCPS. It is focused on applying the Estimation Maximization (EM) algorithm, an approximation scheme to ML, to vehicle parameter estimation. This section is based on the results of the student thesis.

The Maximum Likelihood estimation problem is formulated as follows: A nonlinear,

discrete-time system with the unknown parameters θ, the state x, the differential equation f_θ^d, the process noise w, the measurable output y, the measurement function h_θ and the measurement noise v is considered. The discrete time differential equation f_θ^d can be derived from the continuous time function f_θ, by applying an integration scheme such as Runge-Kutta. To simplify the notiation, the index d is dropped in the following and f_θ is used for both continuous and discrete time functions, depending on the context. The index θ is used to indicate that both f_θ and h_θ depend on the values of the parameter vector θ.

$$x_{i+1} = f_\theta(x_i, u_i) + w_i \tag{56}$$

$$y_i = h_\theta(x_i, u_i) + v_i \tag{57}$$

The likelihood function $\mathcal{L}_\theta(y_0, y_1, \cdots, y_N)$ is defined as the log probability density function of the random variables $[y_0, \cdots, y_N]$ for a given parameter vector θ, (e.g. how likely is a given measurement sequence if a parameter vector θ is assumed).

$$\mathcal{L}_\theta(y_{0:N}) = \log\left(p_\theta(y_{0:N})\right) \tag{58}$$

The maximum likelihood parameter θ, which best explains a given measurement sequence $[y_0, \cdots, y_N]$ is:

$$\theta^{\mathrm{ML}} = \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}_\theta(y_{0:N}) \tag{59}$$

If the parameter θ is considered a random variable, the posterior distribution can be formulated with Bayes' rule. The terms $p(\theta)$ and $p(y_{0:N})$ are usually (at least here) considered unimportant for the estimation task, so that:

$$p(\,\theta|\,y_{0:N}) = \frac{p_\theta(y_{0:N}) \cdot p(\theta)}{p(y_{0:N})} \propto p_\theta(y_{0:N}) \tag{60}$$

For arbitrary distributions of v and w, as well as for nonlinear f, there is possibly no simple closed-form solution for $\mathcal{L}$. Therefore the ML problem is reformulated in the EM algorithm [34]. Using a distribution function for the hidden states, $q(x_{0:N})$, a function $F_\theta[q(x_{0:N})]$ is defined,

$$F_\theta[q(x_{0:N})] := \int q(x_{0:N}) \log \frac{p_\theta(x_{0:N}, y_{0:N})}{q(x_{0:N})} \mathrm{d}x_{0:N} \leq \log\ p_\theta(y_{0:N}), \tag{61}$$

which integrates over all possible sequences of hidden states, to achieve a log-likelihood estimate for the parameter θ, (lower bound for the true probability). According to [31], the function F

Algorithm 1 Abstract Expectation-Maximization algorithm [31, 34]

1: **procedure** EM0(θ^0)
2: **for** $k = 0, 1, \cdots$ **do**
3: $\quad q^{k+1} \leftarrow \underset{q(x_{0:N})}{\operatorname{argmax}} F_{\theta^k}[q(x_{0:N})]$ *(E-Step)*
4: $\quad \theta^{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} F_{\theta}[q^{k+1}]$ *(M-Step)*
5: **return** θ_k

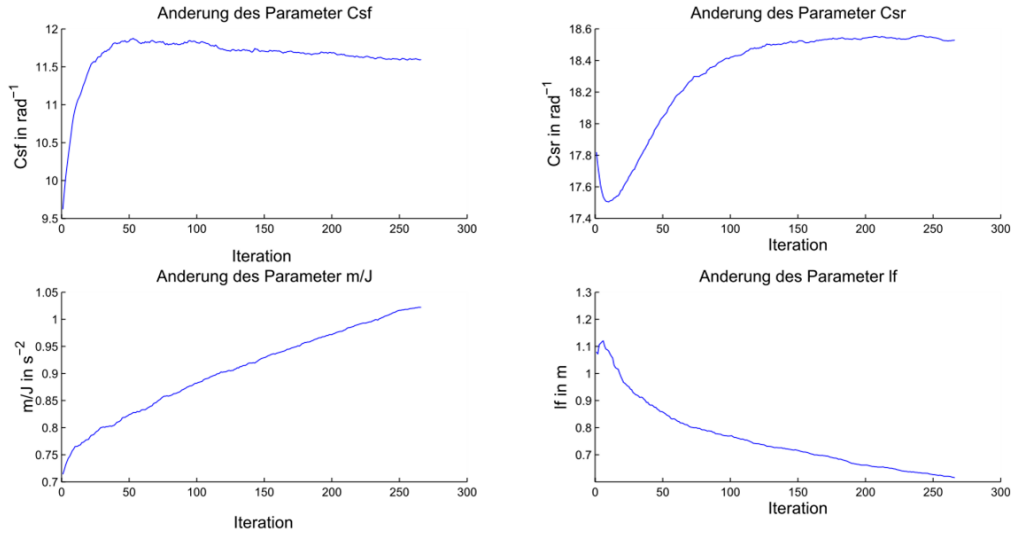

Figure 17: Development of the parameters using the iterative EM-algorithm. The x-axis indicates iterations of the algorithm.

can be maximized by sequentially maximizing first the hidden state distribution q and then the parameter θ. The concrete implementation of the algorithm follows the derivations in [34] and makes use of a particle smoother in the estimation step, [38]. The EM-algorithm is applied to a 5 second long subset of the testdrive given above, in order to find suitable parameters. The results of the iterative parameter computation of the EM-algorithm are shown in Fig. 17. The results of the identification with the EM-method are given in Tab. 6. One can observe considerable differences between the estimates θ^{LS} and θ^{EM}. This might be due to the different lengths of the training sets and will be further investigated.

4.5 Representation of Reference Trajectories in UnCoVerCPS

In order to exchange plans between the cooperating vehicles in work package 5.3, the movement of the vehicle over time has to be represented. We will use the same format as a set value

$\frac{I_z}{m}$	$\frac{b}{L}$	c_f	c_r	(h)	(L)	(g)	(μ)
0.98	0.77	-11.59	-18.53	0.5	2.7	9.81	1

Table 6: Estimated vehicle parameters θ^{EM}. (The parameters in brackets are not evaluated and are taken as a given.)

(an input) to the tracking controller, which realizes the plan. The reference trajectory (plan) has to specify a coordinate for the vehicle position for a future point of time on the interval $[t_0, t_0+T]$, with t_0 being the start time and T the duration of the reference trajectory. In order to be usable for the controller, the reference trajectory has to specify smooth acceleration changes (according to the relative degree of the vehicle model). Several function types could be employed, which satisfy this property, e.g. polynomials, B-splines, sinusoidals or spirals of fitting degree. In the following, spiral functions of degree 1 will be used, due to their minimal number of parameters. To check and to guarantee compatibility of the vehicle plans, a maximum deviation for all vehicle body parts from the reference trajectory has to be specified. Tab. 7 describes the message form, which represents spiral reference trajectories: The parameters of a first order spiral are computed as follows from the message data:

$$\kappa(t_0) = \mathrm{V0}^2/\mathrm{Rvv0} \tag{62}$$

$$\dot{\kappa} = \mathrm{V0}^2/\mathrm{dRvv} \tag{63}$$

Fig. 18 displays the representation of the trajectory and space which is reserved by such a trajectory. In the example, the values $T = 4s$, $\kappa_0 = 1/40m$, $\dot{\kappa} = -1/25m/10m$, $W = 2m$, $L = 3m$ are used.

4.6 Multivariate prediction control model

For the definition of an autonomous tracker, able to follow a predefined trajectory, a multivariate predictive control model [42] is used, which is widely applied in autonomous driving applications [5, 12, 13, 14, 26, 27, 28, 36, 41].

It allows to take advantage of previsualized road information, and thus allows to obtain the optimal control action leading to a minimisation of the future error relative to the executed

Name	Unit	Size (bytes)	Type	Description
type	enum	1	unsigned byte	type of function
t0	0.01 s	8	unsigned long	start time in UTC
T	0.01 s	2	unsigned short	duration
X0	0.01 m	8	unsigned long	UTM easting
Y0	0.01 m	8	unsigned long	UTM northing
PSI0	0.01 °	4	unsigned int	heading, ccw from east
Rvv0	0.1 m/s^2	4	signed int	start lateral acc.
dRvv	0.01m/s^2	4	signed int	change of Rvv0
V0	0.01 m/s	4	unsigned int	initial tangential velocity
dV0	0.01 m/s^2	4	signed int	change of V0
W	0.01 m	4	signed int	width of covered space
L	0.01 m	4	signed int	length of covered space

Table 7: Parameters for the description of a set trajectory

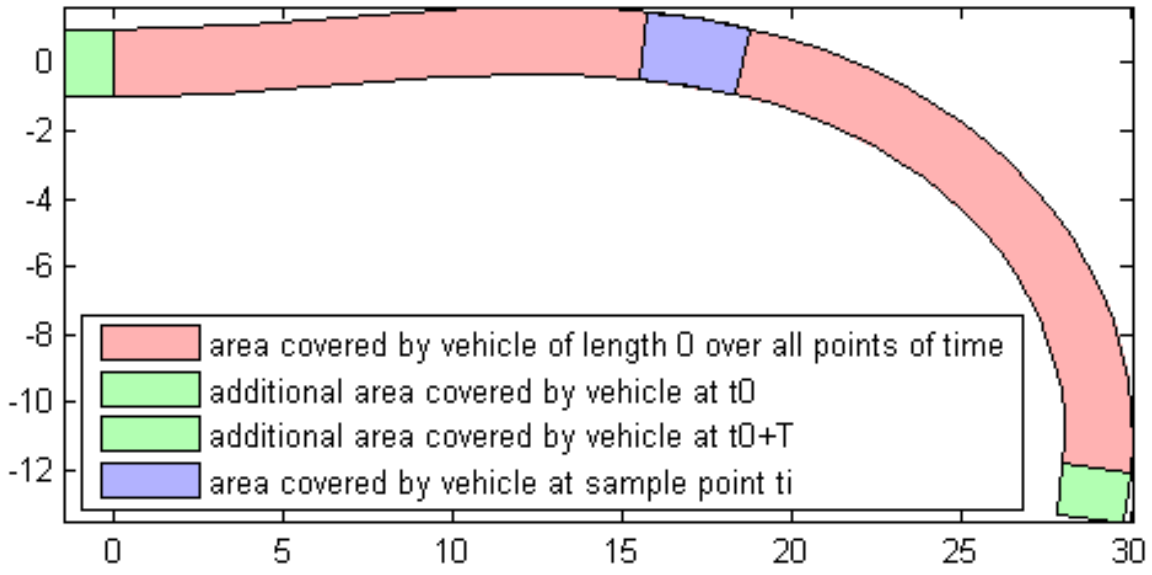

Figure 18: Example for a trajectory represented by a first order spiral and the space possibly covered by the vehicle.

trajectory. Based on the discretized system described in (64)

$$\begin{aligned} x(t+1) &= A_c x(t) + B_c u(t) \\ y(t+1) &= C_c x(t+1) \end{aligned} \tag{64}$$

where $x(t) = [x_1(t)..x_n(t)]^T$ is the state of the system, $u(t) = [u_1(t)..u_{n_u}(t)]^T$ represents the input of the system and $y(t) = [y_1(t)..y_{n_y}(t)]^T$ is the output of the system. A_c, B_c and C_c are the discretized state space matrices. Using a prediction horizon H_p, we can describe the system state in successive time frames:

$$\begin{aligned} \widehat{x}(t+1|t) &= Ax(t) + Bu(t) \\ \widehat{x}(t+2|t) &= A\widehat{x}(t+1) + Bu(t+1) \\ &= A^2x(t) + ABx(t) + Bu(t+1) \\ &... \\ \widehat{x}(t+H_p-1|t) &= A\widehat{x}(t+H_p-1) + Bu(t+H_p-1) \\ &= A^{H_p}x(t) + A^{H_p-1}Bu(t) + A^{H_p-2}Bu(t) + ... + Bu(t+H_p-1) \end{aligned} \tag{65}$$

With the objective of guaranteeing a smooth movement of the steering wheel in the automated vehicle, a smooth transtion between system states has to be ensured. Therefore, we optimize the variation of the control signal $\Delta u(t)$ instead of the control signal itself. This way, the state prediction will be represented by incremental control inputs:

$$\begin{aligned} \Delta u(t) &= u(t) - u(t-1) \\ u(t) &= \Delta u(t) + u(t-1) \end{aligned} \tag{66}$$

The predicted output $\hat{y}$ is obtained by replacing (66) in (65) and rearranging,

$$\hat{y} = Fx(t) + G'u(t-1) + G\Delta u = f + G\Delta u \tag{67}$$

For the optimization of the lateral position on the road, a cost function is defined that penalizes any deviations from the reference path, being the reference signal the center line of the desired path. The cost funtion is defined as follows:

$$J(H_p, H_u) = (\hat{y} - w)^T Q((\hat{y} - w) + (\Delta u(t+k-1|t)^T R \Delta u(t+k-1|t)^T), \tag{68}$$

where $w = [w(t+1)...w(t+H_p)]^T$ determines the future system references in the prediction timeframe $t = k$ with $k \in t + H_p$. The matrices Q and R are defined as the diagonalization of the weighting factors for the trajectory deviations and the weighting factors for the control, respectively:

$$Q = \begin{bmatrix} q & 0 & ... & 0 \\ 0 & q & ... & 0 \\ 0 & 0 & ... & q \end{bmatrix}, R = \begin{bmatrix} r & 0 & ... & 0 \\ 0 & r & ... & 0 \\ 0 & 0 & ... & r \end{bmatrix}.$$

Deriving the optimization of the cost function in (68) we obtain the control law for the optimal future control signal Δu that best follows the reference path. When no constraints are taken into account, an analytical solution for the minimal cost function can be found:

$$\Delta u = (G^T Q G + R)^- 1 G^T Q (w - f) \tag{69}$$

For the case in which restrictions are introduced, it is necessary to solve the equation with successive iterations (i.e. with the Matlab *quadprog* command).

4.7 I/O-Linearization-based tracking controller

In the following, a feedback formula for the steering angle and the desired longitudinal acceleration is derived, using Input-Output Linearization according to [23]. We assume that a set trajectory is given, which specifies a desired position $r(t)$. The path heading θ and its derivatives, as well as the desired tangential velocity v_{ref} and its derivative are required to compute the controller error and the feedback:

$$\theta = \tan^{-1}\left(\frac{\dot{r}_y}{\dot{r}_x}\right) \tag{70}$$

$$v_{ref} = \sqrt{\dot{r}_y^2 + \dot{r}_x^2} \tag{71}$$

$$\dot{\theta} = \kappa v_{ref} \tag{72}$$

$$\ddot{\theta} = \frac{\partial \kappa}{\partial s} v_{ref}^2 + \kappa \dot{v}_{ref} \tag{73}$$

In the following, a rotation matrix $R(x) = \begin{pmatrix} \cos(x) & -\sin(x) \\ \sin(x) & \cos(x) \end{pmatrix}$ is used. A reference point p is selected for the vehicle, which is a distance l_x ahead of the vehicle center of gravity c, $p = c + R(\psi)[l_x, 0]^{\mathrm{T}}$. The controlled output vector y is defined as the longitudinal and lateral deviation of the look-ahead point from the set-trajectory given in set trajectory coordinates:

$$y := R(-\theta)\,(p - r)\,. \tag{74}$$

The input vector u is defined as:

$$u := [a_x, \delta]. \tag{75}$$

By computing first and second derivatives of y, one can determine that the vehicle's relative degree is two and that u directly influences $\ddot{y}$. To realize a certain desired feedback, e.g. $\ddot{y}^* := -k_0 y - k_1 \dot{y}$, the acceleration of the vehicle's center of gravity a_{xy}, as well as its rotational acceleration $\dot{\omega}$ have to fulfill:

$$R(\psi - \theta) \begin{pmatrix} -\ddot{\theta} y_1 - \dot{\theta}^2 y_0 - 2\dot{\theta}\dot{y}_1 + \ddot{y}_0 + \dot{v}_{ref} \\ \ddot{\theta} y_0 - \dot{\theta}^2 y_1 + 2\dot{\theta}\dot{y}_0 + \ddot{y}_1 + \dot{\theta} v_{ref} \end{pmatrix} = a_{xy} + l_x \begin{pmatrix} -\omega^2 \\ \dot{\omega} \end{pmatrix}. \tag{76}$$

The required longitudinal acceleration directly follows. Using the inverse of the vehicle dynamics model, the input steering angle can be computed from the required lateral and rotational acceleration:

$$a_y + l_x \dot{\omega} = k_f \alpha_f + k_r \alpha_r \tag{77}$$

$$\Rightarrow \delta := \tan^{-1}(v_y + (L-b)\omega, v_x) - \alpha_f, \tag{78}$$

using

$$\alpha_f = \tan^{-1}(v_y + (L-b)\omega, v_x), \tag{79}$$

$$\alpha_r = \tan^{-1}(v_y - b\omega, v_x), \tag{80}$$

$$k_f = \left(1 + l_x (L-b) \frac{m}{I_z}\right) \left(\frac{b}{L} g - \frac{h}{L} a_x\right) \mu c_f, \tag{81}$$

$$k_r = \left(1 - l_x b \frac{m}{I_z}\right) \left(\frac{L-b}{L} g + \frac{h}{L} a_x\right) \mu c_r. \tag{82}$$

4.8 Vehicle to vehicle communication model

This section aims to describe the communication model between the UnCoVerCPS vehicles, that will share the trajectory of the manoeuvre before it is executed. This communication model and implementation will increase the driving safety since each vehicle will be aware of the trajectory of any other vehicle before its execution.

Lane change cases

Vehicles shall implement a common communication protocol for negotiation of cooperative maneuvers and this protocol shall define roles for cooperating cars. In the next figure

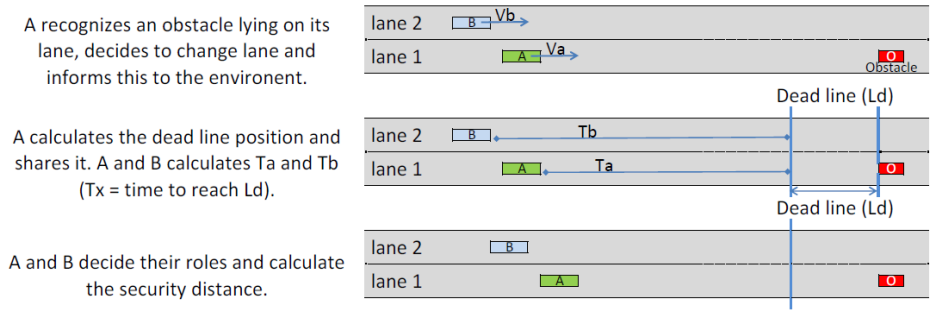

Figure 19: Vehicle to vehicle starting situation

a lane change situation is explained engaging communication between two vehicles. The communication protocol is based on the master and slave role assignment between the vehicles. The vehicle which would like to perform a maneuver will be the master (vehicle A) and the other vehicles involved in this maneuver will be the slaves (vehicle B). A trajectory negotiation

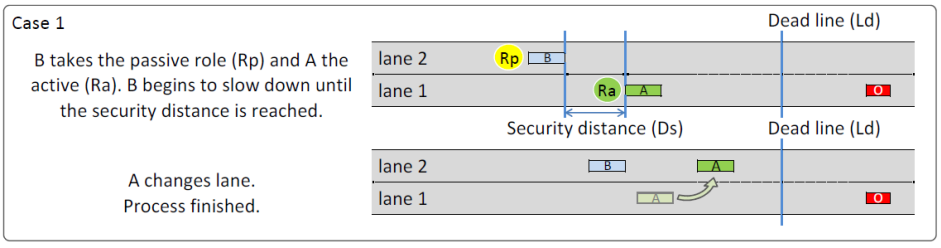

Figure 20: Vehicle to vehicle lane change case 1

is required between the two involved vehicles, in order to guarantee free space for maneuvres and free occupancy of intersectional regions. Figures 20 and 21 represent two different use cases for lane changes, where trajectory negotiation between vehicles must be considered.

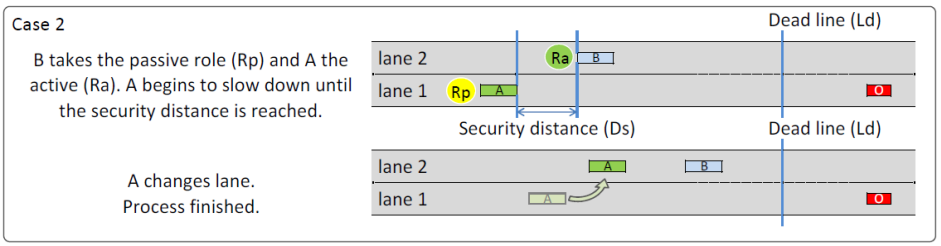

Figure 21: Vehicle to vehicle lane change case 2

Lane change protocol

The lane change maneuver is supported by the communication protocol illustrated in Fig. 22 and Fig. 23. The diagrams show the messages transmitted between two vehicles in case of fragmentation (22) and without fragmentation (23), summarizing the three phases and messages described below. The message fragmentation mechanism is used if the payload data size exceeds the maximal payload size, in order to limit the maximal message size to avoid errors. The process of lane changing starts when a vehicle A decides to initiate a lane change manoeuvre. The communication protocol implements three phases:

- Action plan negotiation: the master and slaves negotiate their manoeuvres to agree a common action plan.
- Waiting action plan start: the master and slaves are waiting to start performing the common action plan
- Action plan execution: the master and slaves start to perform the common action plan.

Vehicle A calculates a manoeuvre and transmits it to vehicle B with a $PROPOSAL$ message. Vehicle B receiving the $PROPOSAL$ message, assumes the slave role. Acknowledgeable (ACK) messages shall always be acknowledged by the receiving party. After receiving a $PROPOSAL$ message, the slave vehicle tries to design a compliant manoeuvre for the proposed manoeuvre. If a vehicle fails to design a compliant manoeuvre it sends a $PROPOSAL\_REJECTED$ message. If a vehicle accepts a proposed manoeuvre it sends a $PROPOSAL\_ACCEPTED$ message with its manoeuvre. When vehicle A receives the $PROPOSAL\_ACCEPTED$ message, both cars execute the manoeuvres and the lane change process ends. A vehicle sends $MVR\_FINISHED$ messages to each session partner after finishing successfully its planned manoeuvre. Any vehicle must send an ABORT message after detecting any unexpected event that could jeopardize the planned execution of the plan of action. There are 10 different messages that vehicles can send to each other for communication:

- $PROPOSAL$: source vehicle broadcasts this message with its proposed trajectory.
- $PROPOSAL\_ACCEPTED$: Destination vehicle accepts source vehicle's trajectory and broadcasts this message with its proposed trajectory. Source vehicle accepts destination vehicle's trajectory and broadcasts this message without any proposed trajectory.

- $PROPOSAL\_REJECTED$: destination vehicle rejects source vehicle's trajectory.
- $COUNTERPROPOSAL$: destination vehicle rejects source vehicle's trajectory but proposes its own trajectory as a new trajectory proposal to be considered by the source vehicle for generating a compliant source vehicle trajectory. This is an explicitly desired redundant message for avoiding errors in communication
- $ABORT$: a vehicle aborts its manoeuvre.
- ACK: it is sent by a vehicle to acknowledge previous $PROPOSAL$, $PROPOSAL\_ACCEPTED$, $PROPOSAL\_REJECTED$ or $COUNTERPROPOSAL$ messages.
- $HEARTBEAT\_PREPROCESSING$: it is sent by a vehicle waiting to start the manoeuvre.
- $HEARTBEAT\_PROCESSING$: it is sent by a vehicle while it is performing the manoeuvre.
- $THINKING$: it is sent by a vehicle while processing or generating a compliant trajectory.
- $MVR\_FINISHED$: it is send by a vehicle when it has finished its manoeuvre.

The $PROPOSAL$, $PROPOSAL\_ACCEPTED$, $PROPOSAL\_REJECTED$, $ABORT$ and $MVR\_FINISHED$ messages are acknowledgeable messages and require an ACK message from the receiver.

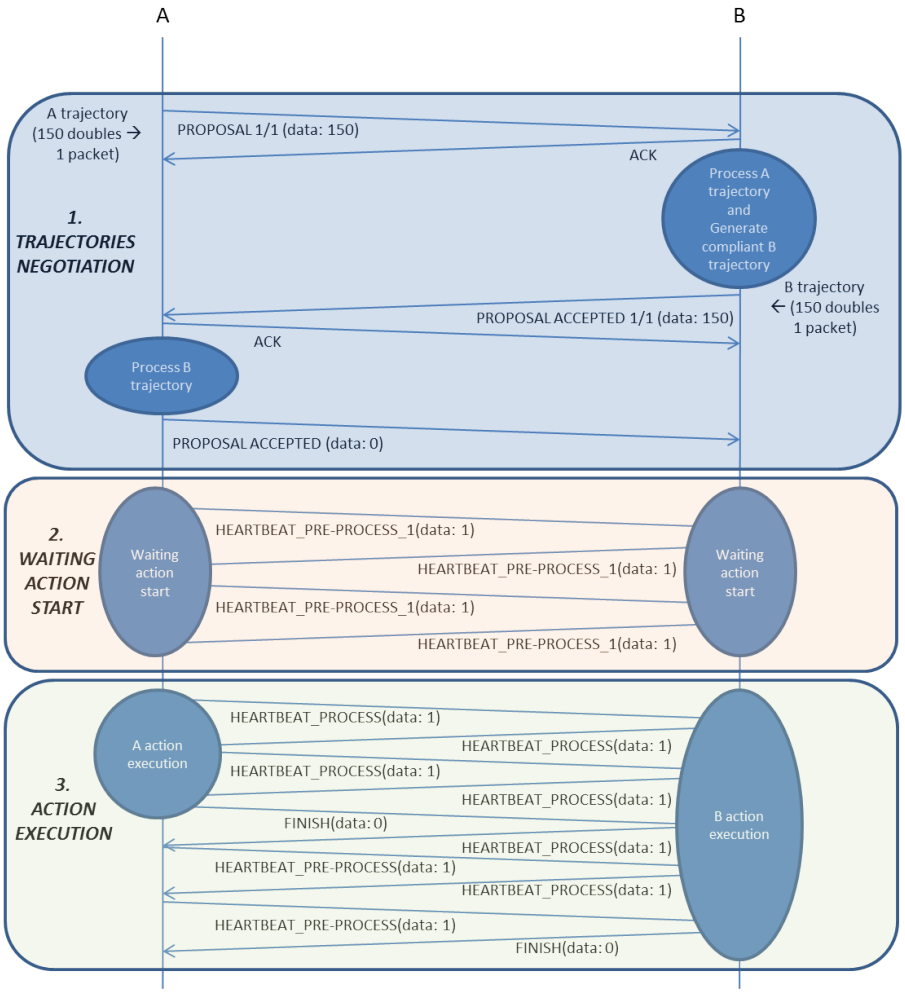

Figure 22: Vehicle to vehicle lane change protocol 1

It is important to define time constraints in order to know if the communication failed or any message is lost. The clocks will be synchronized using the GPS clock.

- The maximal waiting time for an ACK message is 20ms: when an acknowledgeable message arrives, an ACK message shall be sent to the sender by the receiver within the maximal waiting time. If an ACK message exceeds the timeout, the message being acknowledged shall be considered lost and shall be resent.
- During the pre-processing phase all vehicles shall broadcast $HEARTBEAT\_PREPROCESSING$ messages every 100ms.
- During the processing phase all vehicles shall broadcast $HEARTBEAT\_PROCESSING$ messages every 100ms.

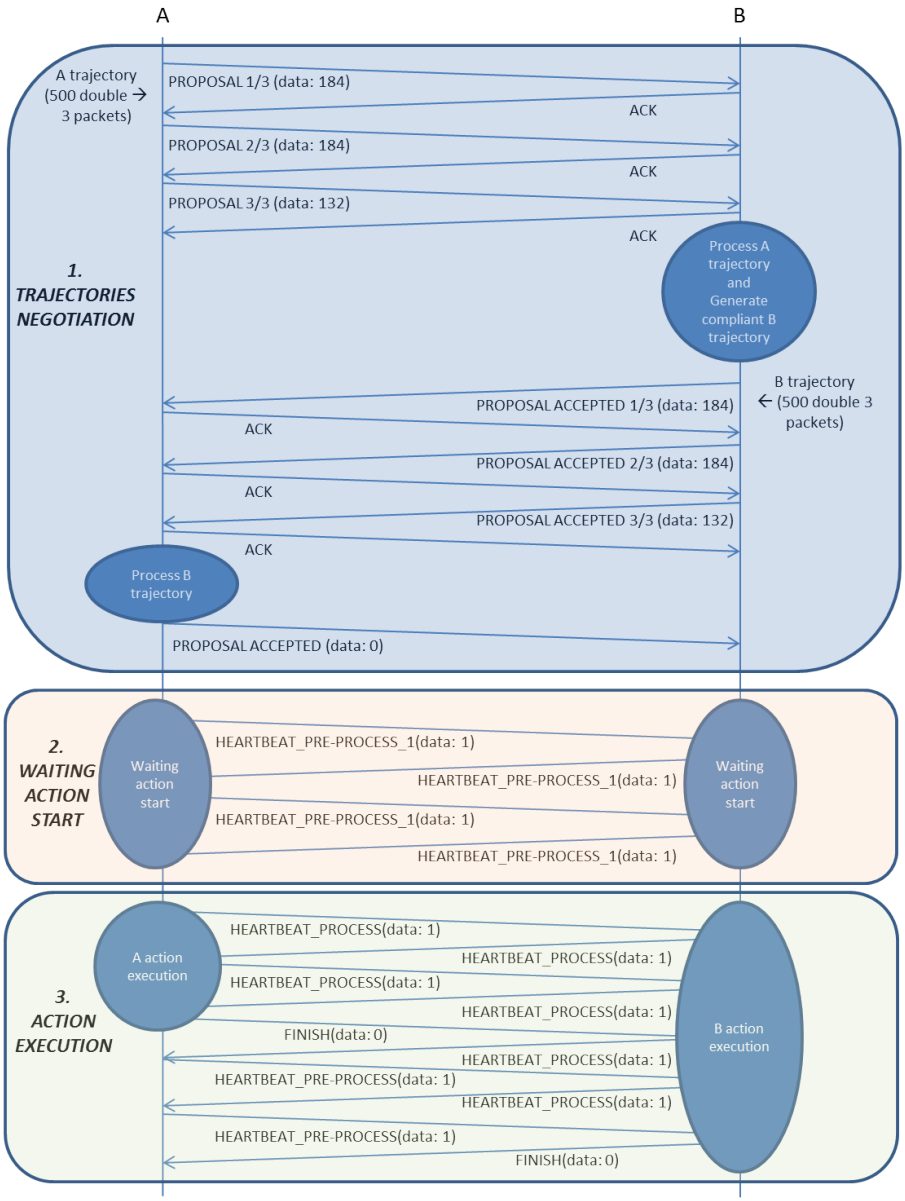

Figure 23: Vehicle to vehicle lane change protocol 2

State Machine

Based on the defined communication protocol as presented in (22) and (23), a high-level state machine of the vehicle to vehicle communication model has been developed. Figure 24 presents the implemented state machine, where basically the relationship between the different possible states related to all the messages defined in section 4.8 are presented. Some important remarks to clarify the initiation and end of a session are:

- The vehicle wanting to perform a cooperative manoeuvre initiates a new session.
- The vehicle initiating a session assumes the master role for that session.
- The vehicle receiving a $PROPOSAL$ message assumes the slave role for that session.
- The successful delivery of a $PROPOSAL$ message determines the beginning of a session.
- The end of a session is determined by the successful delivery of a:
  - $PROPOSAL\_REJECTED$ message.
  - $ABORT$ message.
  - $MVR\_FINISHED$ message.

This state machine is going to be developed using the SCADE tool and it will be in charge of changing states depending on the messages received from the controller.

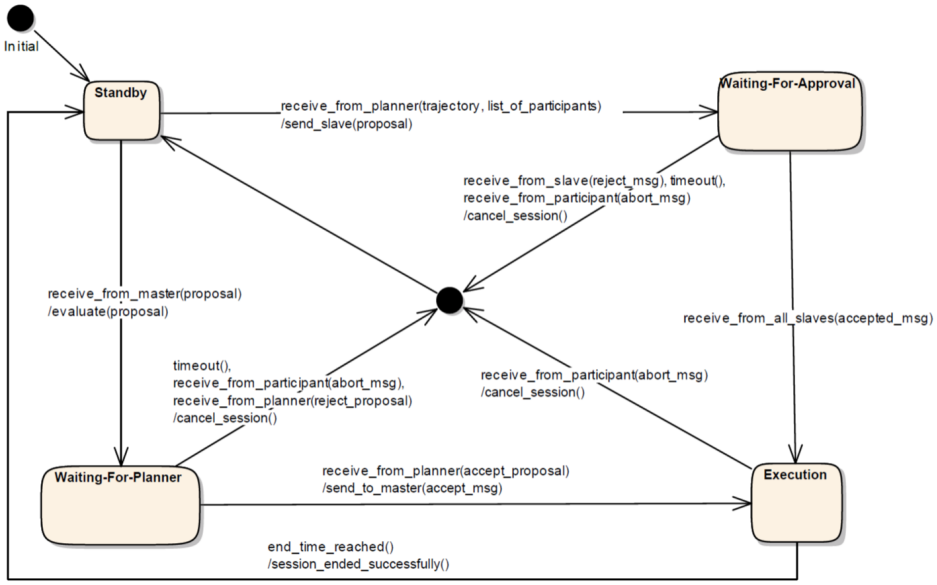

Figure 24: Vehicle to vehicle state machine

5 Model of GRAIL robot in overlapping human and robot workspaces

The GRAIL robot is intended to be used for food assembly tasks carried out in collaboration with human co-workers and other robots. Since most food manufacturing companies are

SMEs, they often cannot afford the large investment necessary to completely automate a production line. The GRAIL robot, therefore, is designed to be deployable alongside other workers (human or robotic) on an incremental basis. The GRAIL robot is shown in Fig. 25.

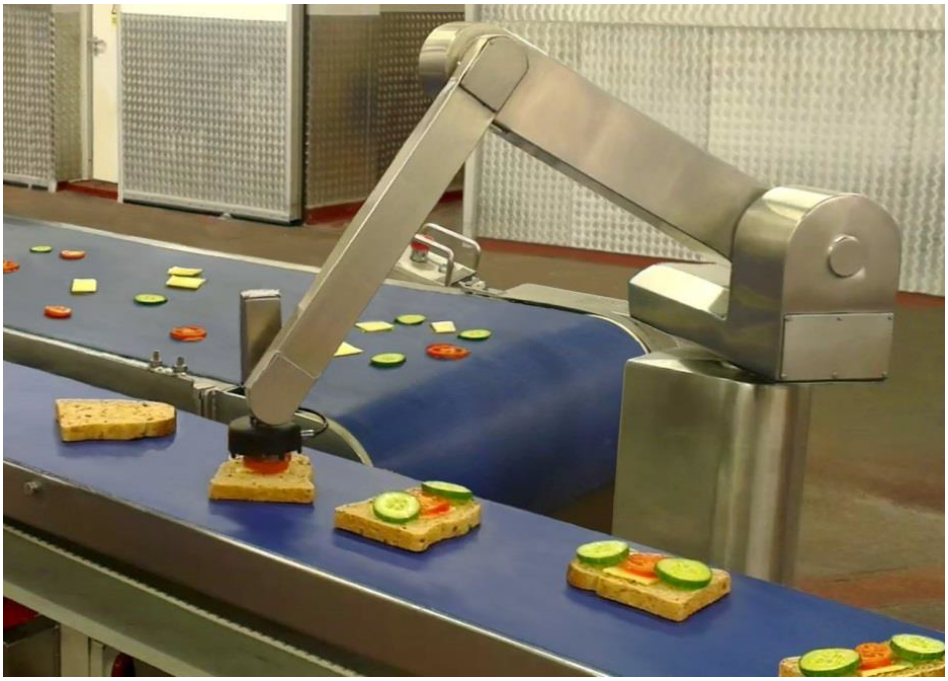

Figure 25: The GRAIL robot arm

The original GRAIL Robot Control System uses only a kinematic model for its control of the GRAIL robot. Low-level control is achieved using a PID control loop that is implemented inside the joint controllers themselves. A dynamic model has not previously been developed and it is not yet clear if a dynamic model will ultimately be required to implement the UnCoVerCPS approach on this system. If required, the dynamic model for the GRAIL robot will be described in deliverable 5.2. The following sections describe the kinematic model of the GRAIL manipulator.

5.1 The kinematic model

The forward and inverse kinematic equations require accurate knowledge of the mechanical relationships between the joints including the link lengths and the angles between the joint axes. Due to the planar kinematic design of the robot, the forward and inverse kinematic equations are quite simple and are defined in Sections 5.2 and 5.3. All terms that are not

defined in the equations relate to dimensions of the arm or frames of reference and are shown in Fig. 26.

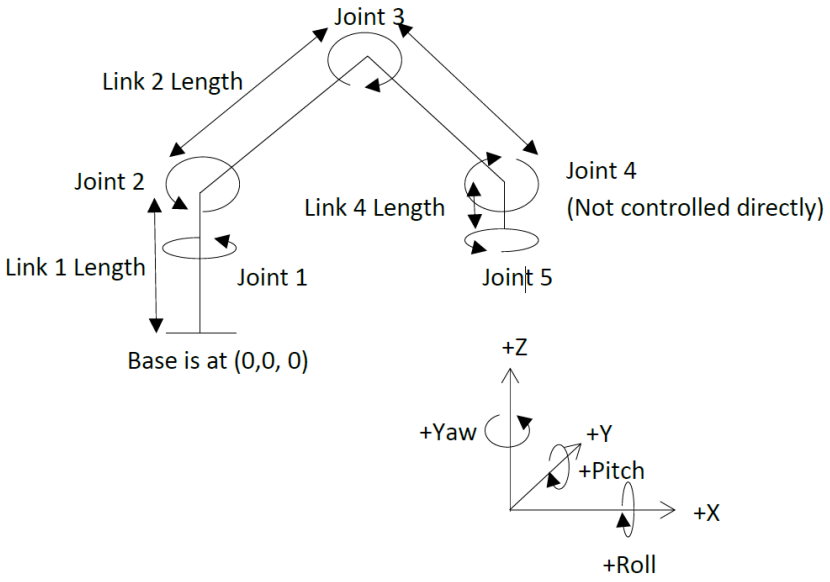

Figure 26: Robot arm kinematic arrangement

Note that joint 4 is mechanically linked to joint 2 and 3 so that link 4 is always vertical, i.e. parallel to Link 1. Joint 4 cannot be controlled independently by the control system. The joints have the limits shown in Table 8.

The position of the arm shown in Fig. 26 is close to the configuration in which all joint angles are zero. When all joint angles are zero, link 2 and link 3 are horizontal and pointing along the positive X axis.

5.2 Forward kinematic equations

The forward kinematic equations relate the robot arm's joint angles to the position and orientation of its end-effector in Cartesian space. The inputs to the forward kinematic equations are the joint angles measured with respect to the zero joint angles defined at the end of Section 5.1 and with the directions shown in Fig. 26. The outputs are the position of the end-effector in terms of its x, y and z coordinates according to the frame of reference shown in Fig. 26, and its orientation in terms of its roll, pitch and yaw angles which are

Joint Number	Minimum Angle (radians)	Maximum Angle (radians)
Joint 1	-3.141592	3.141592
Joint 2	-0.453785	3.403009
Joint 3	-2.879793	2.879793
Joint 4	n/a	n/a
Joint 5	Unlimited	Unlimited

Table 8: GRAIL Robot Joint Limits

rotations about the x, y and z axes, respectively. Some of the equations are expressed in terms of subexpressions which are defined following the main equations. This is done so that the expressions can be given meaningful geometrical interpretations. For example, the term *link2LengthProjectedOntoXY* is the length that link 2 projects onto the xy plane at the given angle of joint 2. Similarly, *link2LengthProjectedOntoZ* is the z-axis component of the length of link 2 at the given angle of joint 2. Once these subexpressions are defined, the derivation of the x, y and z positions of the end-effector is straight forward. The orientation components are similarly simple since roll and pitch are fixed at zero and $\pi/2$, respectively and yaw is determined by the combination of joint1Angle and joint5Angle. Using these descriptive geometric subexpressions and the simple relationships between joint angles and yaw angle, it is clear that the forward kinematics equations can be expressed as:

$$X = (link2LengthProjectedOntoXY + link3LengthProjectedOntoXY) \times cos(joint1Angle) \quad (83)$$

$$Y = (link2LengthProjectedOntoXY + link3LengthProjectedOntoXY) \times sin(joint1Angle) \quad (84)$$

$$Z = link1Length + link2LengthProjectedOntoZ + link3LengthProjectedOntoZ - link4length \quad (85)$$

$$Roll = 0 \quad (86)$$

$$Pitch = \frac{\pi}{2} \quad (87)$$

$$Yaw = joint1Angle + joint5Angle \tag{88}$$

where:

$link2LengthProjectedOntoXY = link2Length \times cos(joint2Angle)$,

$link2LengthProjectedOntoZ = link2Length \times sin(joint2Angle)$,

$link3LengthProjectedOntoXY = link3Length \times cos(joint2Angle–joint3Angle)$,

$link3LengthProjectedOntoZ = link3Length \times sin(joint2Angle–joint3Angle)$

5.3 Inverse kinematic equations

The inverse kinematic equations relate the position and orientation of the robot's end-effector in Cartesian space to the joint angles required to achieve that joint angle. The inputs are the x, y and z position coordinates and the roll, pitch and yaw angles. It can be seen from inspection that the required angle for joint 1 is simply atan2(y, x). Note that atan2 is used rather than a simple arc-tangent so that the correct sign is calculated in all four angular quadrants. It is equally simple to see that the required yaw angle is controlled directly by joint 5. However, joint 1 introduces an additional yaw angle at the end-effector which must be subtracted. The required angle for joint 3 is slightly less easy to calculate. It can be seen from Fig. 27 that a triangle can be formed using the known lengths of link 2 and 3 and the length marked reachLength in the figure which is easy to calculate. The value of Θ is then easy to determine using the cosine rule which results in:

$$cos\Theta = \frac{link2Length^2 + link3Length^2 - reach2Length^2}{2 \times link2Length2 \times link3Length} \tag{89}$$

Since joint3Angle has a simple relationship to Θ, the equation for joint3Angle after rearrangement becomes the equation shown in (92), below.

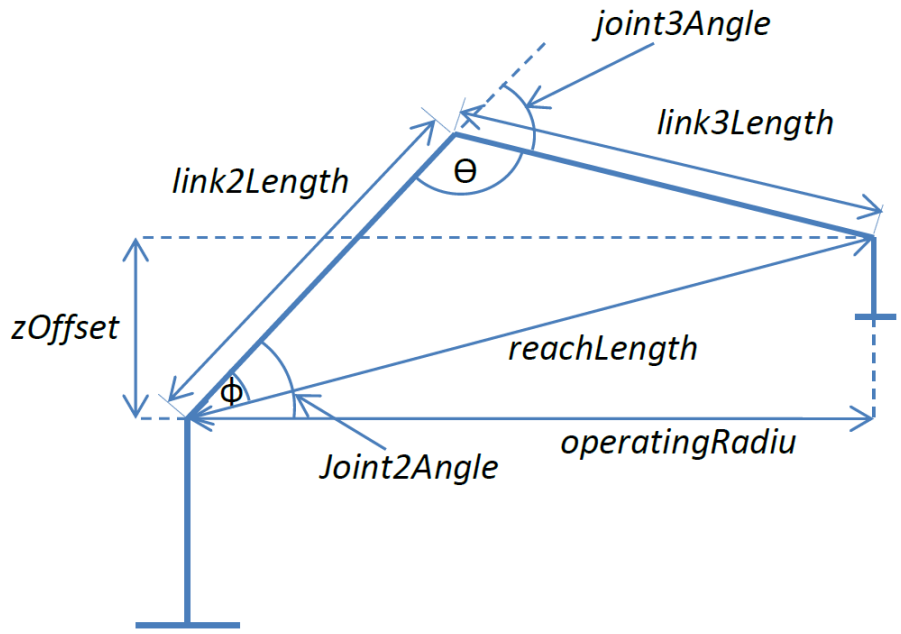

Figure 27: Derivation of joint angle 2 and 3

The angle of joint 2 can be calculated by a similar process to the angle of joint 3 using the same triangle. This time the cosine rule is used to determine ϕ in terms of link2Length, link3Length and reachLength. It can be seen from the figure that joint2Angle is ϕ plus the small angle between reachLength and operatingRadius which can be determined since two lengths are known and they form a right angled triangle. This process results in equation (93) below. Now that all the joint angles have been derived in terms of the input position and orientation, the complete set of equations can be written as follows:

$$joint1Angle = atan2(Y, X) \tag{90}$$

$$\begin{aligned} joint2Angle =& acos\frac{(reachLengthSquared - linkLengthSquaredDifference)}{(2link2Length\sqrt{reachLengthSquared})} \\ &+ atan2(zOffset, \sqrt{operatingRadiusSquared}) \end{aligned} \tag{91}$$

$$joint3Angle = \pi - acos\frac{(linkLengthSquaredSum - reachLengthSquared)}{(2 \times link2Length \times link3Length)} \tag{92}$$

(joint4Angle is not controlled directly by the control system)

$$joint5Angle = yaw - joint1Angle \tag{93}$$

where:

$zOffset = Z + link4length - link1Length$,

$operatingRadiusSquared = (X \times X) + (Y \times Y)$,

$reachLengthSquared = operatingRadiusSquared + (zOffset \times zOffset)$,

$link2LengthSquared = link2Length \times link2Length$,

$link3LengthSquared = link3Length \times link3Length$,

$linkLengthSquaredSum = link2LengthSquared + link3LengthSquared$,

$linkLengthSquaredDifference = link3LengthSquared - link2LengthSquared$

The kinematic design of the robot results in four different possible configurations to achieve most positions and orientations in Cartesian space. These are described as the left-handed/right-handed and elbow-up/elbow-down configurations so that it is possible, for example, have a left-handed, elbow-down configuration. The equations shown are for a right-handed, elbow-up configuration only. Other configurations can be achieved by changing the signs of some of the terms. Practical constraints mean that the right handed, elbow up configuration is the only useful one since other configurations would usually be near joint limits or cause the elbow to collide with the conveyor belt.

6 Summary and Outlook

This report gives a detailed introduction into the use cases considered within UnCoVerCPS, namely wind turbines, smart girds, automated driving and human-robot collaboration. The ultimate goal for the use case selection was to cover a broad range of application areas with completely different verification goals. Within this first deliverable on the use cases, we presented mathematical models in form of dynamical or static equations and, if appropriate, state of the art control algorithms are presented. Further more, the implementation and preliminary simulations results for the different application areas are shown.

The described models are the fist step towards the application of the verification and validation methods under development within UnCoVerCPS. Going forward, these models will be used to test, validate and compare the novel verification methods developed in UnCoVerCPS. The models will also serve as formal mathematical description for the implementation for the demonstration purposes.

Appendix

A Implementation and preliminary results

At the date of this report, several preliminary implementations have been developed and first simulations have been conducted for the wind turbine, the smart grid, the automated driving, and the GRAIL robot applications. Since they are very application dependent, they will be described separately in the following sections of this Appendix.

A.1 Wind turbine

The model as described in Section 2 is implemented in Matlab/Simulink, using the parametrisation of the reference turbine [25]. The NREL turbine has a rated power output of $P_r = 5$MW, a rated rotor speed of $\Omega = 12.1$rmp at a rated wind speed of $v_r = 11.2$m/s. The following files are available for the consortium partners:

Filename	Description
`SimplifiedWTModel.slx`	Simulink model of the wind turbine and controller
`SimplifiedTurbine_Config.m`	configuration file, define here 'SingleRun' and wind speed 'URef' or 'AllCases' for all wind speeds
`SimplifiedTurbine_Main.m`	run this file in Matlab to simulate the turbine
`SimplifiedTurbine_ParamterFile.m`	specifies all turbine and controller data
`SimplifiedTurbine_PostProcessing.m`	load calculation and other post processing
`aeromaps.mat`	maps for c_P and c_T as polynomial functions
`InitialConditions.mat`	initial conditions for the integrators
`wind/ClassA_config.mat`	configuration for the TurbSim input files and effective wind speed calculation
`wind/ClassA.mat`	three random seeds per wind speed for $v_0 = 4 : 2 : 24$m/s according to IEC Ed. 3, Class A

To calculate the damage equivalent loads, the WAFO toolbox [39] is needed.

Simulations of 630s were performed. The first 30s of each simulation are dismissed and the remaining 600s are used for the analysis. Pitch actuators are limited to $0° \leq \theta_c \leq 45°$ and a rate of $\dot{\theta} = 8°/\text{s}$. Generator torque is limited to $M_{g,c} \leq 15000.0\text{Nm}$ and torque rate to $\dot{M}_g \leq 47402.91\text{Nm/s}$.

Figure 28 and Figure 29 show simulation results for below-rated wind speed (mean wind speed 4m/s and turbulence intensity 34%) and above-rated wind speed (mean wind speed 14m/s and turbulence intensity 18%).

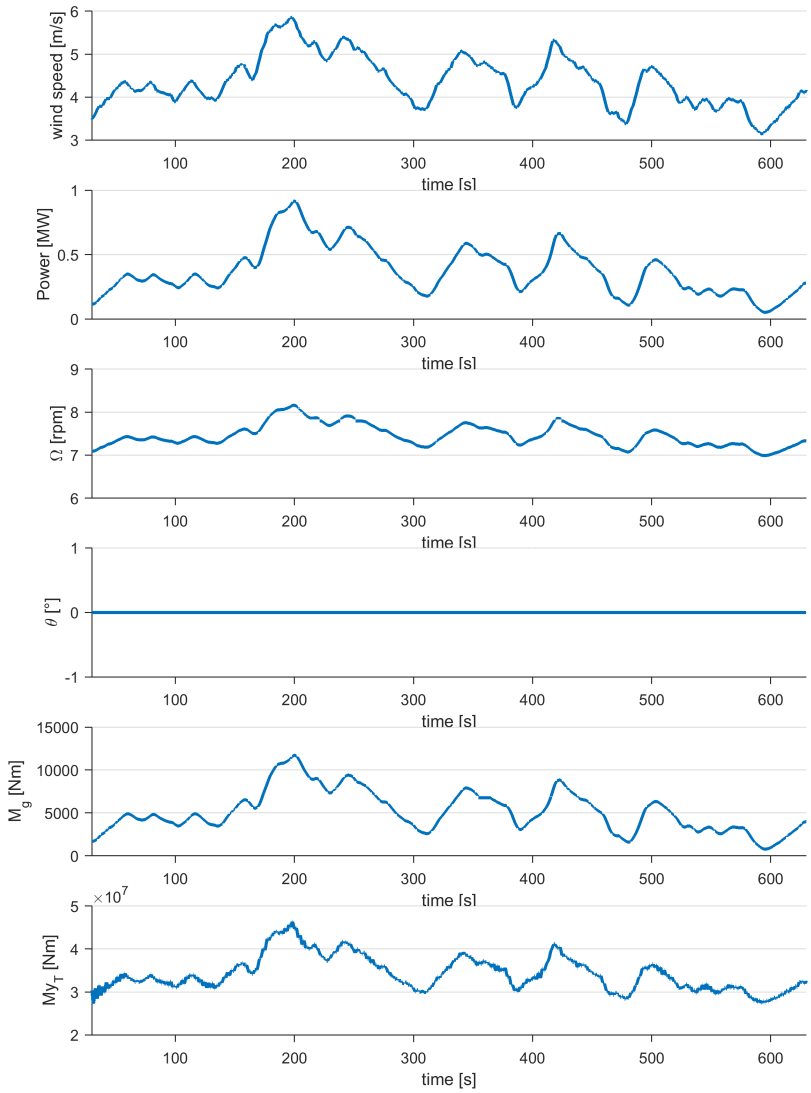

Figure 28: Simulation results for a mean wind speed of 4m/s and a turbulence intensity of 34%.

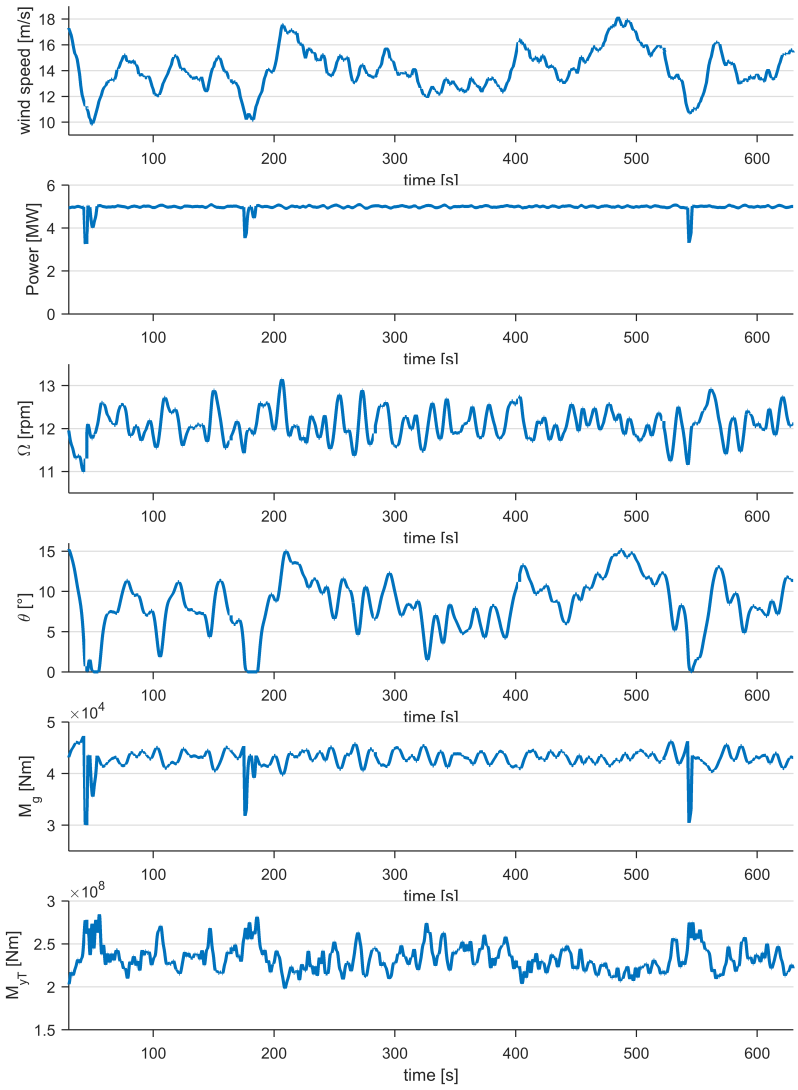

Figure 29: Simulation results for a mean wind speed of 14m/s and a turbulence intensity of 18%.

While the pitch actuator is inactive below rated wind it is active above rated wind to control the rotor speed. Generator torque is varying below-rated wind speed, but almost constant above rated wind. Note that due to the high turbulence intensity, we also enter region 2, even though the mean wind speed above rated wind speed. This can also be seen at the power output in Figure 29, that is almost constant whenever the turbine operates above rated speed and drops for lower wind speeds.

We also run a full load case with three seeds for wind speeds $v_0 = 4 : 2 : 24\text{m/s}$ to calculate the damage equivalent loads. The results are summarized in Figure 30.

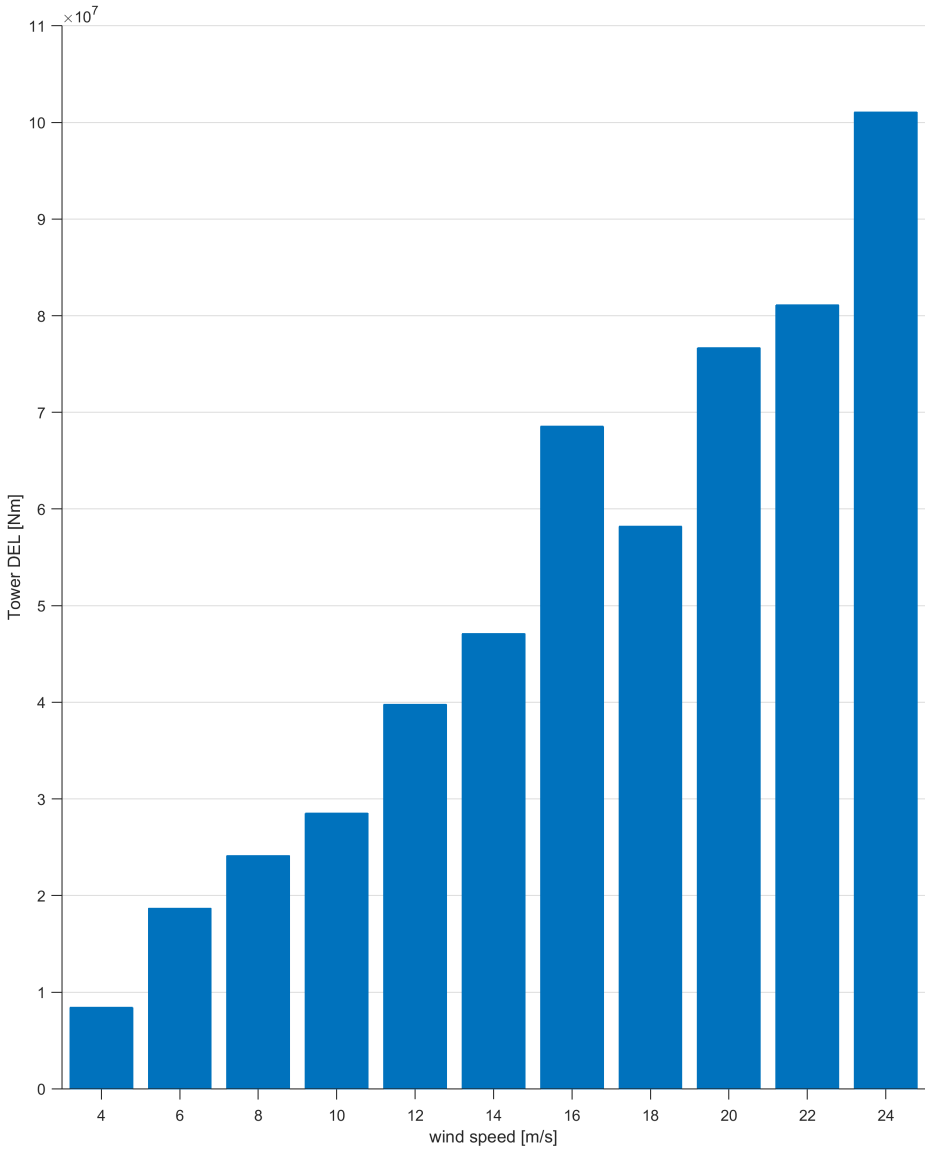

Figure 30: Simulation results for tower damage equivalent loads, three seeds per wind speed.

A.2 Smart grid

In this section we show the results of some simulations obtained from a Matlab implementation of the smart grid configuration in Figure 10.

The smart grid operates in connection with the main grid, that can accomodate for electrical energy requests to compensate demand and generation mismatches within the smart grid. The control variables are set so as to minimize the costs of the electrical energy request to the main grid, while guaranteeing the satisfaction of users energy needs. As for the disturbances entering the smart grid description, a certainty equivalence based control design strategy is adopted with reference to some nominal profile. More specifically, the problem of minimizing the electrical energy cost while guaranteeing comfort and accounting for physical limits is formalized as the following convex optimization problem in the control inputs $\boldsymbol{u}$ (zone temperatures) and $\boldsymbol{s}$ (storage exchanges):

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{s}} \quad & \phi \cdot E_{ch,l}(\boldsymbol{u},\boldsymbol{s}) \\ \text{subject to: } & \boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max}, \quad |\boldsymbol{s}| \leq s_{max} \\ & 0 \leq \boldsymbol{S} \leq S_{max}, \quad E_{ch,l} \geq 0, \quad E_{ch,l}(\boldsymbol{u},\boldsymbol{s}) \leq E_{max} \end{aligned} \tag{94}$$

where $\phi = [\phi(1), \cdots, \phi(M)]$, $\phi(k)$ being the electricity cost in time slot k, and $E_{ch,l} = [E_{ch,l}(1), \cdots, E_{ch,l}(M)]^\top$, $E_{ch,l}(k)$ being the electrical energy requested by the chiller to produce the cooling energy $E_{ch,c}(k)$ in time slot k. $E_{ch,l}(k)$ is computed from $E_{ch,c}(k)$ via (31). In turn, $E_{ch,c}(k)$ is given by

$$E_{ch,c}(k) = E_{r,c}(k) - s(k)$$

where $E_{r,c}(k)$ is the cooling energy request in (11) (which depends on the zone temperatures) and $s(k)$ it the cooling energy provided by the storage. Obviously, $E_{ch,c}$ depends on $\boldsymbol{u}$ and $\boldsymbol{s}$ and this dependence is here made explicit in (94) via the notation $E_{ch,c}(\boldsymbol{u},\boldsymbol{s})$.
As for the comfort and actuation constraints in (94), E_{max} is the maximum amount of electrical energy that the chiller can draw, S_{max} is the maximum amount of energy that the thermal storage can contain, s_{max} is the maximum amount of energy that can be stored/retrieved in/from the thermal storage in a single time slot, $\boldsymbol{u}_{min}$ and $\boldsymbol{u}_{max}$ are vectors representing temperatures lower/upper bounds for all time slots.

The numerical results illustrated next refer to a medium-sized office building which is 20 m long, 20 m wide, and 10 m tall. The building is divided into three floors, each facade is half glazed and the roof is flat. In the following we will consider a single-zone setup, where the three floors are treated as a single zone with the same temperature set-point, and a multi-zone

setup, where each floor is a zone with its own temperature set-point. The look-ahead time horizon is discretized in $\Delta t = 10$ minutes time slots and is 48 hours long, though the strategy is then applied over a one-day time horizon. This is to avoid the depletion of the storage at the end of the day. A thermal storage with maximum capacity $S_{max} = 700$ MJ, a maximum charge/discharge rate of $s_{max} = 18$ MJ within each time slot, and a loss coefficient $a = 0.99$ is considered. As for the chiller, we set $E_{max} = 30$ MJ, and $c_1 = 1.1133 \cdot 10^{-5}$, $c_2 = 1.85 \cdot 10^{-2}$ and $c_3 = 3.6837$ in (31).

We consider nominal realistic profiles for the external disturbances. Such profiles are depicted in Figure 31 for the first 24 hours. For the next 24 hours we consider the same profiles. In Figure 32 we report the upper and lower bounds for the zones temperature (blue solid lines) and the profile of the energy price (green dashed line) during the first 24 hours. The period from 8 AM to 5 PM is referred to as "working hours". As for all other parameters, the reader is referred to [20].

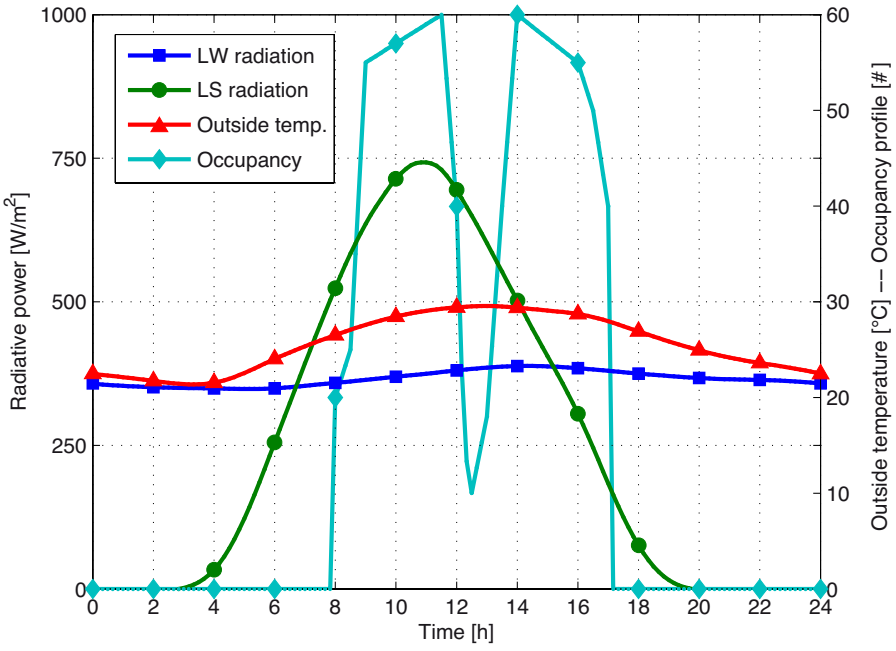

Figure 31: External disturbances profiles.

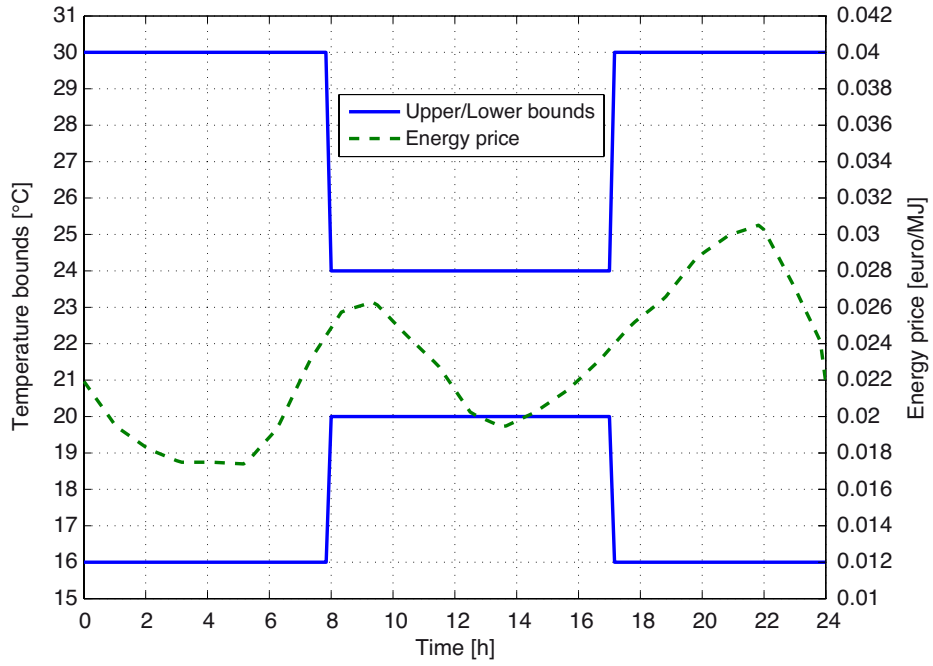

Figure 32: Temperature bounds and energy price profile.

Single-zone setup: Two different strategies are compared: *Fixed*, i.e., during working hours the temperature is maintained at $24°C$, while during the rest of the time the chiller is idle; and *Optimal*, i.e., temperature profiles are determined by solving (94).

For each strategy we consider two cases: with and without thermal storage. In the fixed strategy, the thermal storage is charged during the night and discharged during working hours.

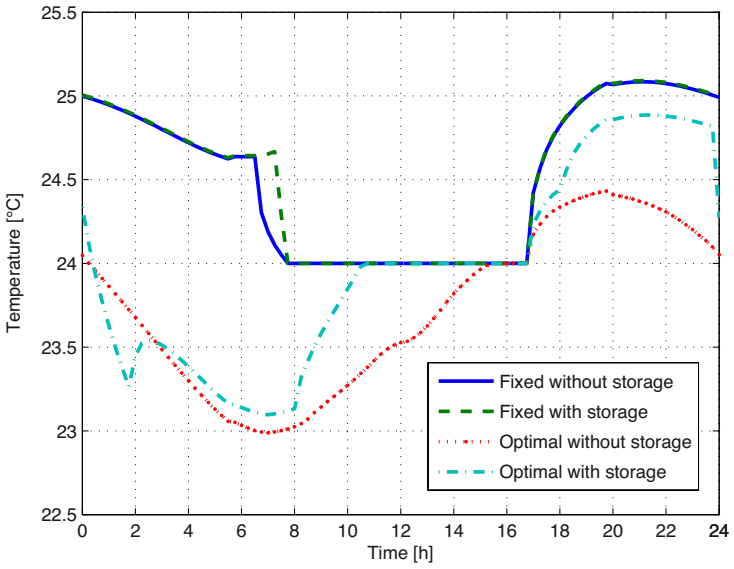

Figure 33: Temperature profiles comparison.

In Figure 33 the zone temperature profile over one period is reported, when different strategies are applied. Almost no difference can be noticed between the fixed strategy with thermal storage (green dashed line) and the same strategy without it (blue solid line). Although both optimal strategies (with and without thermal storage, cyan dot-dashed line and red dotted line respectively) provide different temperature profiles, we can identify three common phases: a pre-cooling phase, where the building temperature is lowered before working hours, a comfort phase in which the temperature is kept within the prescribed limits, and a final phase, where the temperature rises until a pre-cooling phase starts over again.

Figure 34 plots the profile of the energy requested to the chiller. A substantial difference can be seen between the fixed strategy without thermal storage and the other strategies. The proposed control strategy is able to compensate the lack of a thermal storage exhibiting an energy request similar to the fixed strategy case with thermal storage. This is achieved by exploiting the thermal inertia of the building structure as a storage, as suggested in [19]. In the case of the optimal control strategy with storage capabilities, the availability of both active and passive thermal storages allows the chiller to work closer to its best efficiency point. This is confirmed by the results in Table 9, where the overall energy needed for cooling, the energy requested from the chiller, the electrical energy consumption, and the total energy cost for

all strategies (F = fixed without storage, F+S = fixed with storage, O = optimal without storage, O+S = optimal with storage) is reported. The optimal policy (with storage) uses only a little more than a quarter of the storage capacity (see Figure 35). These considerations suggest the use of smaller chiller and thermal storage unit.

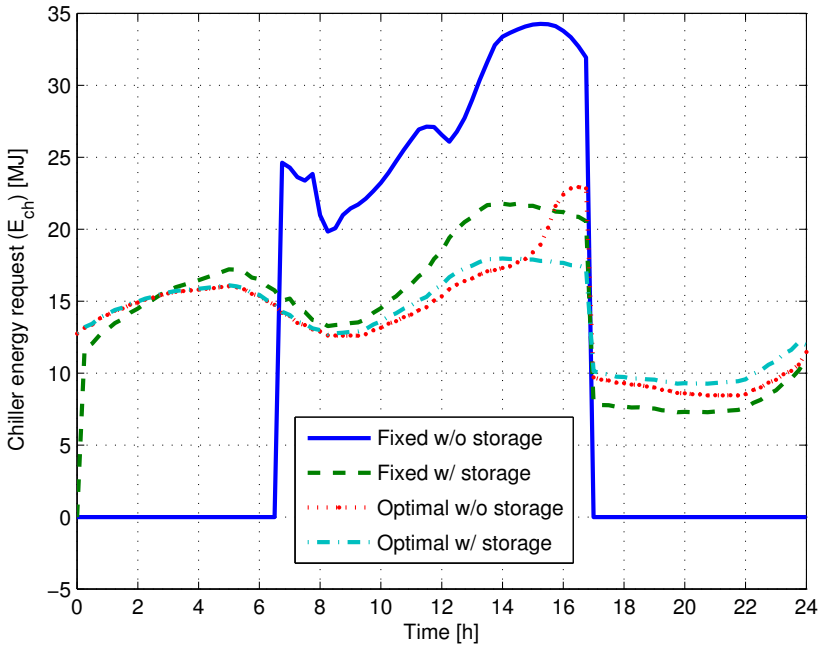

Figure 34: Chiller energy request profiles comparison.

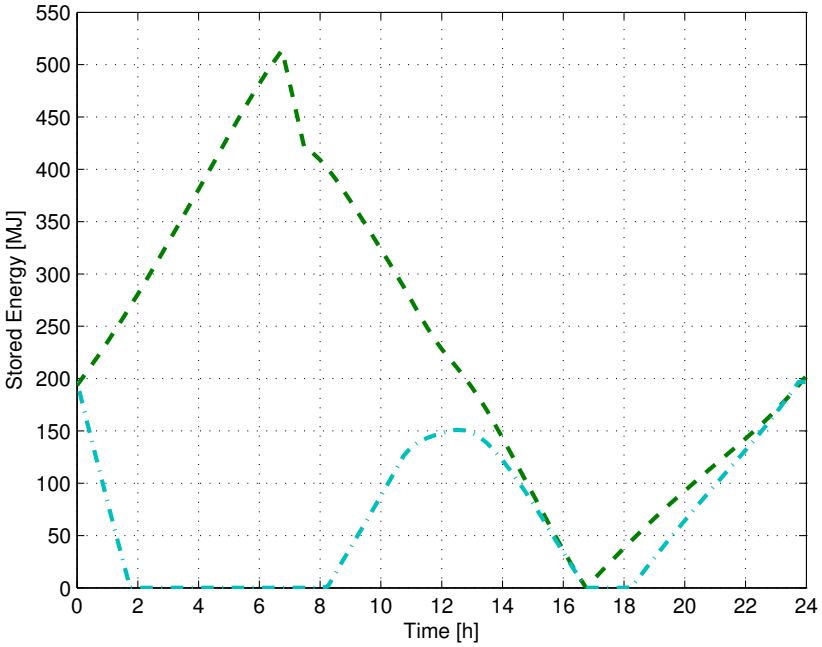

Figure 35: Stored energy profiles comparison.

Strategy	F	F+S	O	O+S
E_c [MJ]	1094	1087	1288	1076
E_{ch} [MJ]	1094	1330	1288	1187
E_ℓ [MJ]	1219	742.3	750.5	694.4
Cost [euro]	29.09	16.95	16.63	14.44

Table 9: Total energy consumption and costs.

Multi-zone setup: Suppose now to adopt a different temperature profile in each floor (ground floor, first floor, and second floor). The optimization procedure can now rely on both thermal inertia of the building structure and thermal exchanges between zones. In the following we focus on the optimal strategy without thermal storage. The resulting temperature profiles depend on the zone and are indeed quite different (see Figure 36). In particular the ground floor (zone 1) is maintained at a temperature level around the upper comfort limit of $24°C$, the second floor (zone 3) follows a temperature profile similar to the single-zone case, and the first floor (zone 2) presents a strong pre-cooling phase in the morning before working

hours, and right after to account for the next day thanks to the 48 hours look ahead horizon.

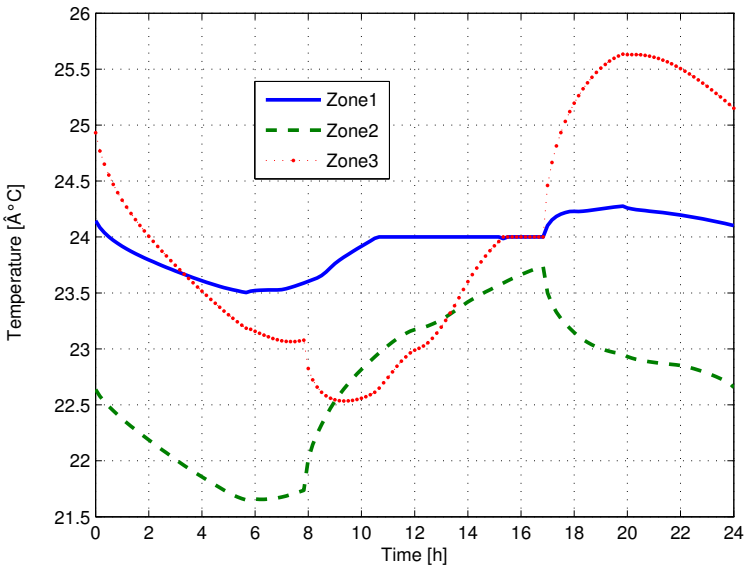

Figure 36: Temperature profiles for each zone.

The first floor is used as a thermal storage which drains heat from other floors through its pavement and its ceiling. Table 10 compares the results obtained with the optimal strategy without thermal storage in the single-zone and multi-zone setups, and shows that using a multi-zone setup can significantly reduce energy consumption and thus cooling costs.

	Single-Zone	Multi-Zone	$\Delta_{\%}$
E_c [MJ]	1288	979.9	-23.9%
E_{ch} [MJ]	750.5	669.7	-14.5%
Cost [euro]	16.63	12.79	-21.1%

Table 10: Single-zone e multi-zone comparison.

A.3 Automated driving

The first implemented controllers towards safety aspects of automated driving are described in the next sections. They include the modelling in SCADE of the tracking controller, as well as preliminary implementations for the vehicle to vehicle communication.

SCADE implementation of tracking controller

The tracking controller model as described in section 4 is implemented in SCADE. Table 11 describes each of the operating blocks of the SCADE model as presented in Figure 38. The regulator computes the current control error and derivative $y, \dot{y}$, and uses the current control gain, the look ahead point l_{cg2p} and a transformation to set trajectory coordinates to compute the required accelerations for the look ahead point, in order to regulate the error to zero. The operator IVM_Bicycle realizes an inverse bicycle model, to compute the correct inputs a_x and δ from the look ahead point acceleration. In case the vehicle's velocity is very low, the output is switched to be generated by the operator LowSpeedController. The operator rk4_clothoid integrates the initial condition of the set trajectory ref_0 over time, in order to produce a reference $r(t)$ for the current point of time. Each time the controller executes, $r(t)$ is integrated over a time interval $[pre(t), t]$. A Runge-Kutta integration scheme of order 4 is used. When the version of the set trajectory (ref_version) changes, the set trajectory reference r is reset to the new ref_0.

SCADE operator	Description
Regulator	Computes acceleration required to regulate tracking error
Normlim	Limits the acceleration to acceptable values
IVM_Bicycle	Computes vehicle inputs from required accelerations
LowSpeedController	Low speed Controller for $v < 5m/s$
rk4_clothoid	Integration of reference trajectory
Tracking_state	Switching between nominal and emergency operation, fig. 37
∗map operators	Convenient mappings from vector inputs to scalar variables

Table 11: SCADE operating blocks for tracking controller

The operator tracking_state keeps track of the controllers state and monitors critical variables, in order to switch to emergency operation, if necessary. The operator is realized as a state machine, as displayed in Figure 37. The state machine has the four states *Inactive, Active, Emergency Inactive* and *Emergency Brake*. The controller starts in Inactive and returns from each other state to Inactive, if the controller is reset. In the state Active, the controller outputs control values in order to regulate the tracking error. The state Emergency

Inactive realizes the case that a human driver wants to take over control. Then, the controller deactivates its outputs in order to give full control to the human. The state is entered when the user presses the brake pedal. The state Emergency Brake realizes the case that an error occurs during tracking: A full braking, no steering maneuver is executed, until the driver takes over control or until the controller is reset. Triggers for the emergency brake can be a delay of the set trajectory messages, $dt_{GPS} + t > t_0 + T$, a GPS timeout (no GPS signal received for a certain amount of time), $dt_{GPS} > GPS\_timeout$, or a fault state indicated by the GPS, $GPS\_state \neq GPS\_ok$ (e.g. not initialized or low precision).

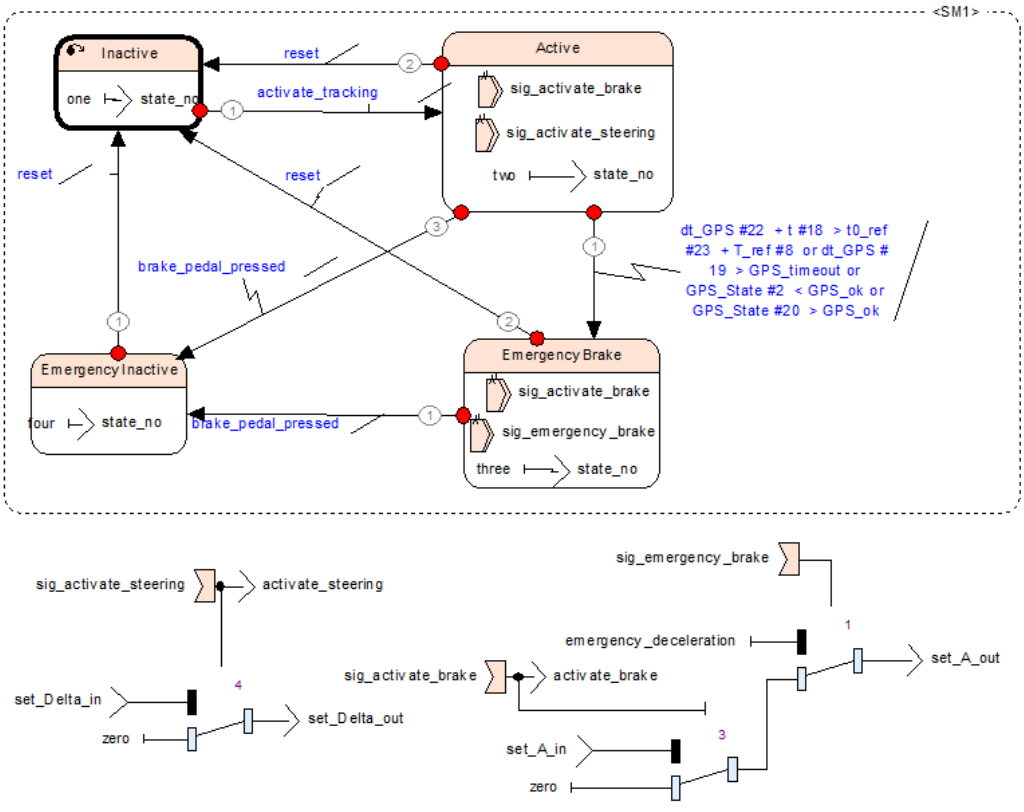

Figure 37: Tracking_state operator: Monitoring for conditions, which require deactivation of control outputs or execution of an emergency braking.

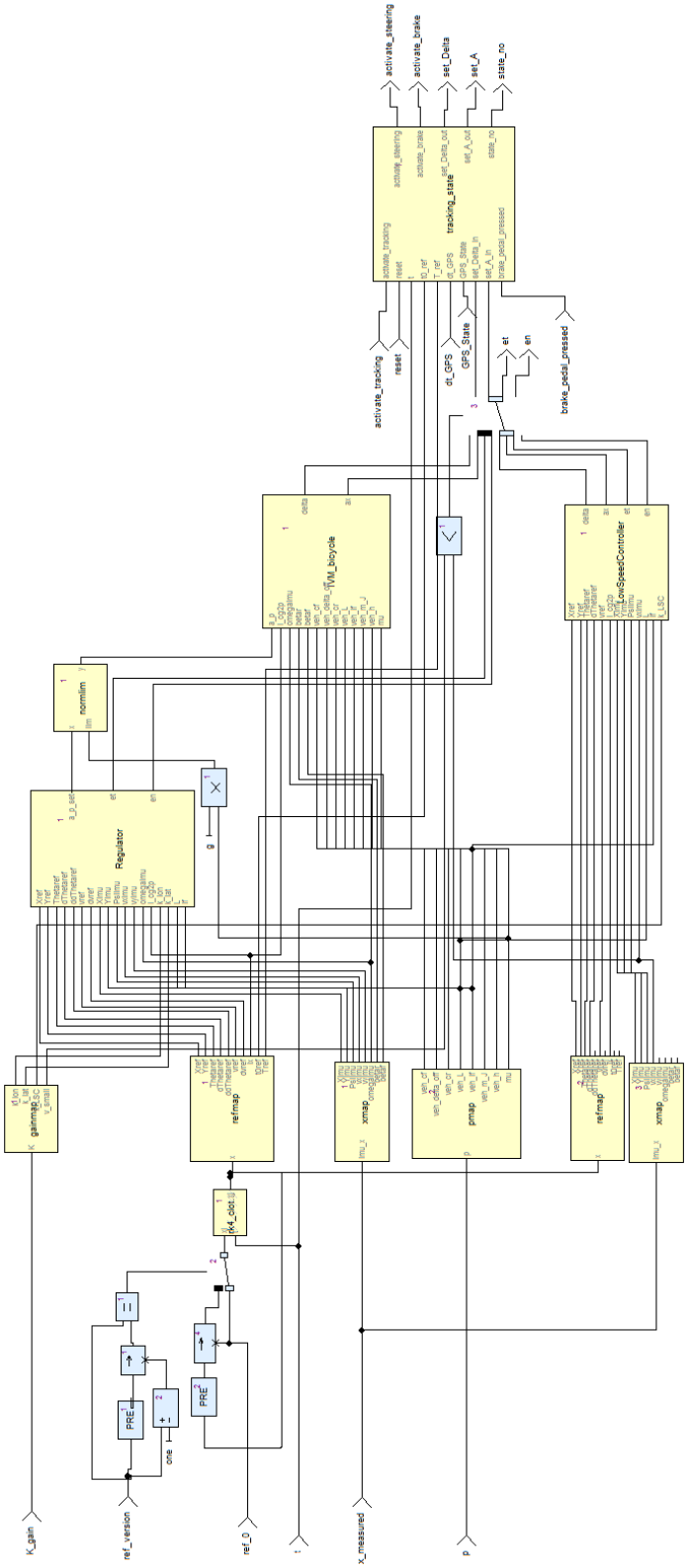

Figure 38: SCADE Tracking Controller implementation

Implementation of vehicle to vehicle communication

Message Structure and Implementation Components The communication protocol uses only one packet format to implement all the different message types that are broadcasted between master and slaves.

The protocol data packet is composed of a 28-byte header and a variable-sized payload. The header is composed at the same time of the following fields:

- $sender\_stationId(64bits)$: identifies uniquely the sender vehicle of a message.
- $receiver\_stationId(64bits)$: identifies uniquely the receiver vehicle of a message.
- $sessionId(16bits)$: identifies uniquely a session. The vehicle initiating a session creates it randomly.
- $messageId(16bits)$: identifies uniquely a session message over a session.
- $partId(4bits)$: identifies uniquely the message part of a fragment message using the MSB 4 bits.
- $partCount(4bits)$: indicates the total count of parts of a fragmented message using the LSB 4 bits.
- $messageType(8bits)$: indicates the type of the message between the following options:
  - $(0)PROPOSAL$
  - $(1)PROPOSAL\_ACCEPTED$
  - $(2)PROPOSAL\_REJECTED$
  - $(3)COUNTERPROPOSAL$
  - $(4)ABORT$
  - $(5)ACK$
  - $(6)HEARTBEAT\_PREPROCESSING$
  - $(7)HEARTBEAT\_PROCESSING$
  - $(8)THINKING$
  - $(9)MVR\_FINISHED$

- $payloadSize(16bits)$: indicates the amount of bytes carried by the payload field.

The payload part of the message contains the maneuver in a PROPOSAL message. This is a cooperative maneuver proposed by the master vehicle. $PROPOSAL\_ACCEPTED$ messages sent by the slave vehicle carry its maneuver in its payload field. $PROPOSAL\_ACCEPTED$ messages sent by the master vehicle contain an empty payload field. The payload field of the $PROPOSAL\_REJECTED$ message is empty.

An ACK message contains the same session and message number as the message being acknowledged.

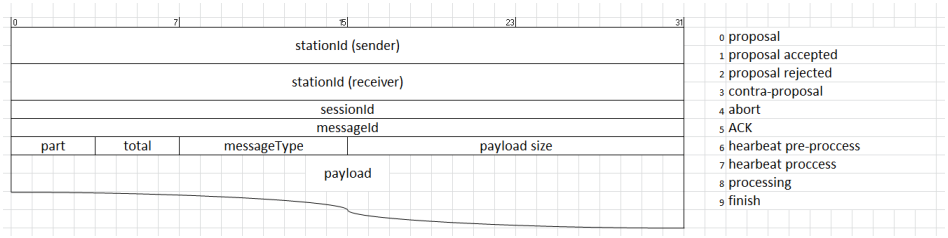

Figure 39: Vehicle to vehicle message structure

After defining all the communication protocol for V2V (vehicle to vehicle), it can be defined some of the components and requirements needed to make this communication possible.

Communication module components The main components of the communication module are presented in Fig. 40:

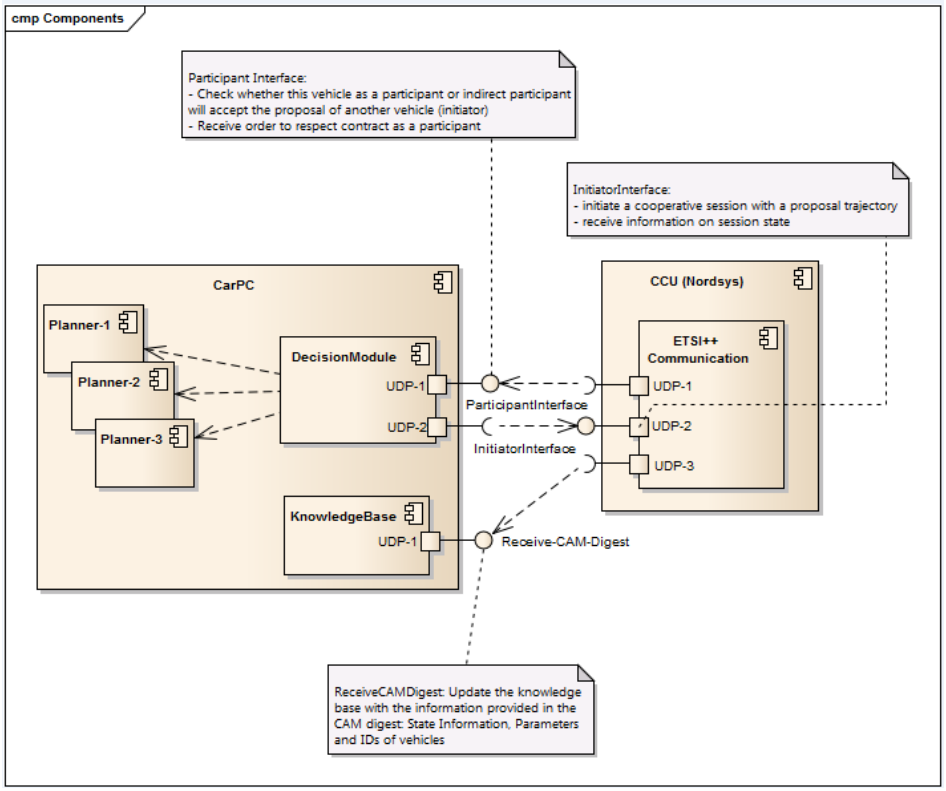

Figure 40: Vehicle to vehicle communication module 1

- *Controller*: this component will receive and send messages to the AdHocCommunication, Dominion and State Machine components.
- *State Machine*: this component will be developed using the SCADE tool and will change from states bearing in mind the messages received from the controller.
- *AdHocCommunication*: this component will receive and send messages to other vehicles using a Nordsys system as shown in Figure 41.
- *Dominion*: this component is in charge of all tasks related to maneuvers.

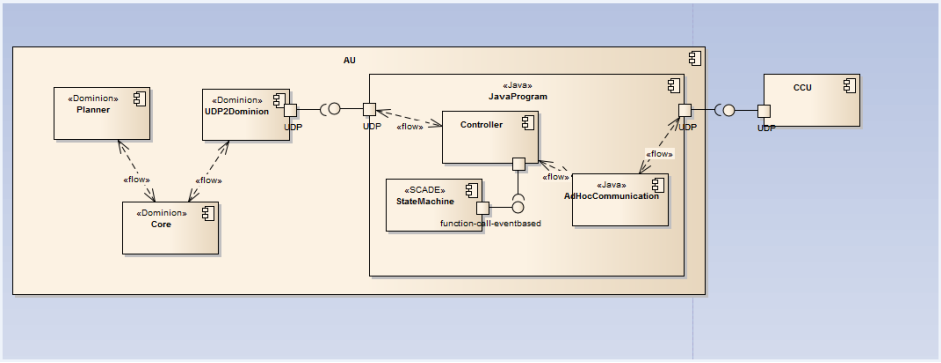

Figure 41: Vehicle to vehicle communication module 2

A.4 GRAIL robot

The implementation of the GRAIL Robot System with the UnCoVerCPS safety system is highly constrained by the safety requirements imposed by legislation. As described in Deliverable D1.1, "Assessment of languages and tools for the automatic formalisation of system requirements", the safety-related parts of the UnCoVerCPS safety system will be classified as a "protective device" and so must conform to performance level "d" (PL=d) as defined in EN ISO 13849-1:2008 [3]. This means that certain safety features must be present in the design. In particular, single faults must not permit an unsafe condition to occur and there must be a means to monitor and detect faults in the safety system before they occur. These legislative requirements apply to a production-ready system. During the research activities of the UnCoVerCPS project, although the system must still be safe, a different approach can be taken. During the research and development phase safety can be maintained through appropriate operating procedures and physical separation as determined by a risk assessment. The researchers will also be more expert in the use of robots and will be familiar with the dangers they create. To achieve PL=d in the production system, the provable correctness of the algorithms used within UnCoVerCPS is not, by itself, sufficient. The hardware and software used to implement the protective device must have an appropriate level of reliability. In practice this means that a formal approach must be taken to the development of the software that includes extensive reviewing and testing. Additionally, the hardware must be designed with fault detection and redundancy in mind. To make this as simple as possible, effort must be made to isolate the UnCoVerCPS "protective device" functionality from the rest of the GRAIL robot control system so that failures in the GRAIL robot control hardware

and software systems (which are not fail safe in their current implementation) do not prevent UnCoVerCPS system from operating. The easiest way to achieve the necessary levels of reliability within the UnCoVerCPS system is through redundancy. This ensures that single points of failure do not disable the safety offered by the protective device. Software bugs can still cause dangerous failures, however, due to the software commonality between the two duplicated systems which is why rigorous processes must be employed to develop this software or diverse algorithms must be developed by independent teams. Unfortunately, although the UnCoVerCPS system can easily be duplicated to ensure the necessary reliability, its output feeds into the GRAIL control system which is in ultimate control of the robot. Failures within the GRAIL controller could, therefore, prevent the safety function of the UnCoVerCPS safety system from operating. This means that any safety critical parts of the GRAIL controller must also be duplicated to maintain safety. An idealized block diagram for a suitable duplicated system appears in Fig. 42. Here, the GRAIL control system has been divided into two parts: the first is responsible for certain high-level activities that do not have a safety impact such as managing the user interface. The second is responsible for all aspects that could cause hazardous operation under fault conditions such as the generation of the trajectories for the end-effector.

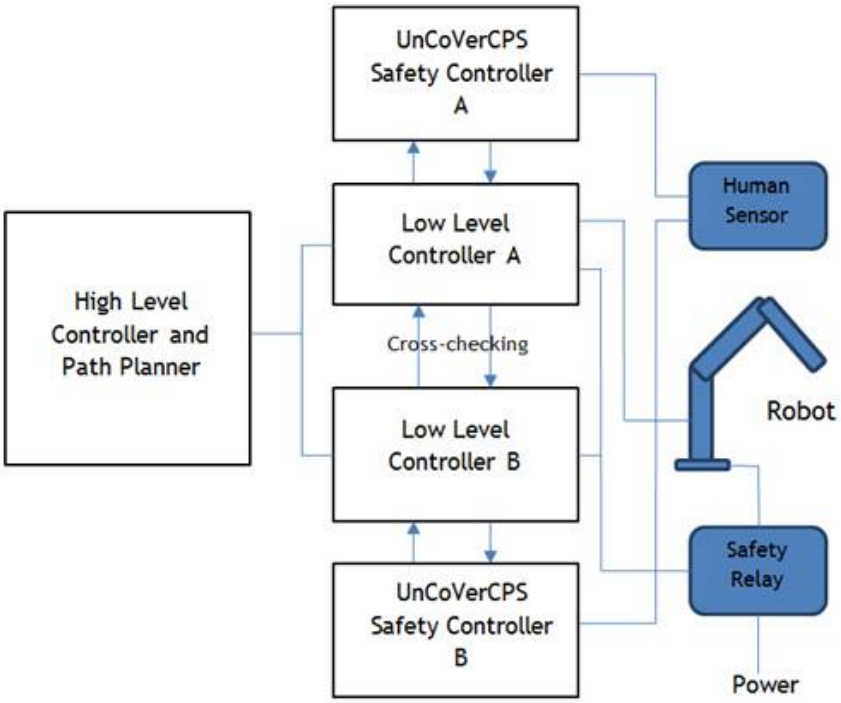

Figure 42: Idealised diagram of control system

In the figure, two UnCoVerCPS safety systems and two safety-related parts of the GRAIL controller are provided. It can be seen that failures in the UnCoVerCPS system or safety-related GRAIL controllers can be detected through the cross-checking between the two systems. In the event that either system believes that the other system is in error a signal to the safety relay causes power to the robot to be cut. When a stop is required due to the safety function then, as long as the two systems agree, a controlled stop can be performed with power still applied to the robot motors. The stationary status of the robot must then be monitored by both of the duplicated channels. Note that only one of the two duplicated systems actually controls the robot. The other is present only so that cross-checking can be performed. The sensor device or devices used to detect the presence of any human co-workers in the robot's workspace also forms part of the "protective device" and so must have suitable reliability. Many potential sensor types are designed for safety applications and so off-the-shelf implementations meet the reliability requirements. However, other more sophisticated sensor types that might be better suited to the application, such as a Microsoft Kinect sensor, do not meet the reliability requirements automatically. Considerable development effort would have

to be expended to design the sensor subsystem so that it always fails safe if such devices were used. A survey of sensor types and simulations currently being performed will determine the sensor type that will be used and will be reported in later deliverables. Table 12 shows how the safety of the system is assured in the presence of various types of failure.

Failure Mode	Protection
Failure of non safety-related parts of the GRAIL control system	Safety is not affected since UnCoVerCPS and safety-related parts of the GRAIL control systems prevent an unsafe situation.
Failure of either UnCoVerCPS A or B (but not both)	Cross-checking causes the safety relay to be opened, cutting power to the robot and resulting in a stop.
Failure of safety-related GRAIL controllers A or B (but not both)	Cross-checking causes the safety relay to be opened, cutting power to the robot and resulting in a stop.
Failure of Human Sensor Output A or B (but not both)	Cross-checking causes the safety relay to be opened, cutting power to the robot and resulting in a stop.
Human encroaches into robot workspace or robot moves towards locations potentially occupied by human	Robot is slowed or stopped as required by UnCoVerCPS A and Low Level Controller A. System B verifies the correctness of the operation. The robot motors remain powered up in this situation, the two duplicated system must continue to monitor for the correct behavior of the robot.

Table 12: Actions on single failure

References

[1] IEC 61400-1 Wind turbines - Part 1: Design requirements, August 2005.

[2] *CIBSE Guide A: Environmental Design*. CIBSE Publications, Norwich, UK, 2006.

[3] *EN ISO 13849-1:2008*. Brussels: CEN, 2008.

[4] ASTM E 1049-85 Standard practices for cycle counting in fatigue analysis, Reapproved 2005.

[5] M. Ali, P. Falcone, and J. Sjöberg. A predictive approach to roadway departure prevention. *IEEE Transactions on Systems*, 14, 2013.

[6] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[7] F. Borghesan, R. Vignali, L. Piroddi, and M.Prandini. Approximate dynamic programming-based control of a building cooling system with thermal storage. In *IEEE ISGT 2013*, Copenhagen, Denmark, October 2013.

[8] E. A. Bossanyi. Individual blade pitch control for load reduction. *Wind energy*, 6(2):119–128, 2003.

[9] C. L. Bottasso, A. Croce, B. Savini, W. Sirchi, and L. Trainelli. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody System Dynamics*, 16(3):291–308, Nov. 2006.

[10] A. Brown. Model-based inclusion of previewed information for lateral vehicle state and environment estimation. *Disertation in Mechanical Engineering*, 2013.

[11] Capstone. *Technical Reference Capstone Model C30 Performance*. Capstone, USA, 2014.

[12] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology, Volume:15 , Issue: 3*, pages 566 – 580, 2007.

[13] P. Falcone, H. Tseng, F. Borrelli, D. Asgari, and J. Hrovat. MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, pages 611 – 628, 2009.

[14] P. Falcone, M. Tufo, F. Borrelli, F. Asgari, and H. Tseng. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans*, pages 2980 – 2985, 2007.

[15] G. Ferrari-Trecate, E. Gallestey, P. Letizia, M. Spedicato, M. Morari, and M. Antoine. Modeling and control of co-generation power plants: A hybrid system approach. In *International Workshop on Hybrid Systems: Computation and Control*, pages 208–224, Stanford, California, USA, mar 2002.

[16] R. Gasch and J. Twele, editors. *Windkraftanlagen - Grundlagen, Entwurf, Planung und Betrieb.* Springer Vieweg, 8th edition, 2013.

[17] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.

[18] M. Gordon and K. Ng. *Cool thermodynamics.* Cambridge International Science Publishing, 2000.

[19] G. P. Henze, C. Felsmann, and G. Knabe. Evaluation of optimal control for active and passive building thermal storage. *International Journal of Thermal Science*, 1(1):191–200, 2003.

[20] D. Ioli. Optimal energy management of a building cooling system with thermal storage: modeling and control. Master's thesis, Politecnico di Milano, Milano, Italy, 2014.

[21] D. Ioli, A. Falsone, and M. Prandini. An iterative scheme to hierarchically structured optimal energy management of a microgrid. In *CDC*, Osaka, Japan, December 2015.

[22] D. Ioli, A. Falsone, and M. Prandini. Optimal energy management of a building cooling system with thermal storage: A convex formulation. In *ADCHEM*, Whistler, British Columbia, Canada, June 2015.

[23] A. Isidori. *Nonlinear control systems*. Springer Science & Business Media, 1995.

[24] J. M. Jonkman and M. L. Buhl. FAST users's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, 2005.

[25] J. M. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 2009.

[26] A. Katriniok and D. Abel. LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics. pages 6828 – 6833, 2011.

[27] S. Keen and D. Cole. Application of time-variant predictive control to modeling driver steering skill. *Vehicle System Dynamics*, 49:527–559, 2011.

[28] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat. Predictive control approach to autonomous vehicle steering. *Proceedings of the American Control Conference*, 15:1 – 6, July 2006.

[29] D. Kim and J. E. Braun. Reduced-order building modeling for application to model-based predictive control. In *5th National Conference of IBPSA-USA*, pages 554–561, Madison, Wisconsin, USA, August 1–3 2012.

[30] D. Kim, W. Zuo, J. E. Braun, and M. Wetter. Comparisons of building system modeling approaches for control system design. In *13th Conference of International Building Performance Simulation Association*, pages 3267–3274, Chambery, France, August 26–28 2013.

[31] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[32] A. D. Platt and M. L. Buhl. WT_perf users guide. Technical Report NREL/TP-XXXXX, National Renewable Energy Laboratory, 2012.

[33] R. Rajamani. *Vehicle Dynamics and Control*. ISBN: 10.1007/978 1-4614-1433-9. Springer, 2 edition, 2012.

[34] S. Särkkä. *Bayesian filtering and smoothing.* Number 3. Cambridge University Press, 2013.

[35] D. Schlipf, D. J. Schlipf, and M. Kühn. Nonlinear model predictive control of wind turbines using LIDAR. *Wind Energy*, 16(7):1107–1129, Oct. 2013.

[36] L. Seung-Hi, O. Young, K. Bo-Ah, and C. Chung. Proximate model predictive control strategy for autonomous vehicle lateral control. American Control Conference, Canada., 2012.

[37] M. Sharp. *Thermal stratification in liquid sensible heat storage.* 1978.

[38] W. Sun. Untersuchung von Algorithmen zur Identifikation der Parameter nichtlinearer Fahrdynamikmodelle. Master's thesis, DLR Braunschweig, TU Braunschweig, 2015.

[39] WAFO Group. WAFO - a Matlab toolbox for analysis of random waves and loads. Technical report, Math. Stat., Center for Math. Sci., Lund University, 2011.

[40] B. Wittenmark, K. J. Åström, and K.-E. Årzén. *Computer Control: An Overview.* Lund University, 2011.

[41] H. Yoshida, S. Shionahara, and M. Nagai. Lane change steering manouvre using model predictive control theory. *Vehicle System Dynamics*, 46, 2008.

[42] A. Zubizarreta. Control predictivo. Master de ingenieria de control, automatizacion y robotica, Universidad del Pais Vasco, 2011.

[43] M. Zurigat and Ghajar. A comparison study of one dimensional models for stratified thermal storage tanks. *Journal of Solar Energy Engineering*, 204-210, 1989.