Online Predictive Control for Daily Irrigation in Open-Field Agriculture

MSc Thesis Systems & Control

Christof Goedhart

Delft University of Technology

Online Predictive Control for Daily Irrigation in Open-Field Agriculture

by

Christof Goedhart

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on March 26, 2026 at 10:00

Student number:	5055210
Project duration:	February, 2025 - March, 2025
Faculty:	Faculty of Mechanical Engineering, Delft

Thesis committee:	M. Guo, PhD	TU Delft ME/DCSC, supervisor
	A. Dabiri, PhD	TU Delft ME/DCSC, chair
	E. Abraham, PhD	TU Delft CEG

TU Delft

Summary

Agriculture accounts for roughly 72% of global freshwater withdrawals, but irrigation is often still based on intuition or fixed schedules. Model-based and data-driven approaches have shown promise, but they mostly rely on historical datasets or crop-specific knowledge that are unavailable in many settings. This thesis addresses that gap by proposing an irrigation control framework that requires no prior data and can learn the system dynamics online during the growing season. The framework combines three components: a Zone Model Predictive Control strategy, an online model estimator based on Recursive Least Squares (RLS), and a Readily Available Water (RAW) estimator. The framework is evaluated using the AquaCrop simulator on a sugarcane case study in southern Mozambique. Results show that the proposed framework performs comparably to a static controller preconfigured with historical data and crop-specific knowledge, while requiring neither. A sparse measurement strategy is also evaluated, reducing the number of required soil moisture measurements by around 65% with little impact on performance. That said, learning the system online comes with trade-offs, and the framework does not consistently outperform a well-configured static model. Together, the results suggest that effective predictive irrigation control is achievable in data-scarce environments.

Contents

1

Introduction

The human population continues to grow and is projected to reach 8.5 billion by 2030 [12]. While the overall growth rate is projected to decline, it will remain high in specific regions of Asia and Africa for the upcoming years. This trend will increase global food demand, underscoring the importance of agricultural practices. In order to optimise crop yield, accurate and sufficient irrigation is essential, as crops lacking a timely water supply will wither, thereby reducing yield.

With the weather becoming more unpredictable due to climate change, including extreme conditions like prolonged droughts and heavy rainfall, rainfall cannot be relied upon to occur at appropriate times [17]. Timely irrigation is required to maintain the soil in an optimal state, where it is neither over-saturated with water nor undersaturated. Smart irrigation involves the precise application of water in appropriate quantities at suitable times and locations [7]. This necessitates monitoring and the implementation of various control strategies to optimise irrigation schedules, taking into account soil characteristics, physiological conditions of the crop, and weather.

Having established the necessity of adequate irrigation in agriculture, challenges remain regarding its efficient and sustainable application. Agriculture accounted for approximately 72% of global freshwater withdrawals in 2021 [2], while over two billion people live in countries with inadequate water supply [24]. This statistic highlights the need to optimise water use to prevent waste and ensure long-term availability. The challenge is finding the right balance between maximising crop yield and not depleting the available freshwater. Challenges also arise in irrigation systems concerning the many factors influencing water requirements, such as variations in soil type, crop characteristics, and weather conditions, creating a dynamic system in which irrigation demands are challenging to predict. As a result, making informed decisions about when and how much to irrigate remains a significant obstacle to achieving optimal water management.

In recent years, different strategies have been studied to improve irrigation practices. Many of these studies take a model-based approach, using a mathematical model of the soil-crop system to support irrigation decisions. The idea is to describe the main dynamics of the soil–plant system well enough to decide when and how much to irrigate, depending on the objective, such as maximising yield, reducing water use, or keeping costs low. These models enable estimation of how specific irrigation schedules or weather conditions will influence the outcome. A commonly used control strategy in this context is

Model Predictive Control (MPC). At the same time, the increasing availability of data has led to more interest in data-driven and AI-based approaches. Instead of relying on a predefined mathematical model, these methods learn the relationship between inputs (e.g., weather and irrigation) and outputs (e.g., soil moisture or yield) from data. Machine learning techniques can capture complex, non-linear behaviour, even when the exact physical relationships are not fully understood.

While previously mentioned model-based and data-driven strategies have demonstrated significant gains over more traditional irrigation methods, successfully implementing these methods can clash with operational practicality and economic reality. One substantial challenge is the reliance on large amounts of data. Model-based techniques require an initial system identification phase that requires a previously collected data set to determine the model parameters. The collected dataset also needs to match the conditions of the problem to be solved; for example, the soil type, crop, and climate might need to match. The conditions and dynamics of growing tomatoes in Spain are quite different from those of growing sugarcane in Mozambique. The same can be said for the previously described deep learning and machine learning approaches, which often require far more data. The assumption is that this data is readily available through Internet of Things (IOT) networks. However, these networks can be very costly, limiting, and a barrier to implementing these approaches. This limitation is especially relevant in countries and regions with fewer economic resources [15]. It raises the question: how can advanced irrigation control methods remain effective when limited data are collected or available? Additionally, some of the proposed control frameworks lack adaptability. One approach is to use a static model, identified before the growing season, to represent the whole system. The model lacks flexibility to adapt to changing conditions and may also misrepresent crop dynamics that evolve over a growing season.

Both model-based and data-driven approaches share a common limitation: they depend heavily on prior knowledge. This may take the form of historical datasets used for model training or system identification, or detailed crop-specific parameters required for seasonal planning. Such dependence often leads to solutions tailored to a specific scenario that are difficult to apply in data-scarce environments or under changing conditions.

There is therefore a need for a system that can function effectively without extensive prior information. Ideally, such a framework can learn the system dynamics online, during the growing season, using only limited measurements. Given these additional constraints, an important question is how much performance can still be achieved. The following research question is formulated:

How can a control framework be designed for daily irrigation scheduling to optimise water use efficiency and prevent crop water stress without relying on prior historical data or crop-specific models, while accommodating practical constraints like sparse soil measurements?

In order to answer this question we will consider the following sub-questions:

1. How can a Model Predictive Control strategy be formulated to maintain soil moisture within optimal ranges while anticipating future weather conditions?
2. How can the dynamic relationship between weather, irrigation, and soil moisture be accurately modelled and updated during the growing season without the use of a priori data?
3. To what extent does the frequency of soil moisture measurements impact the ability of the system to maintain soil moisture within safe limits and optimise water use?

The rest of this thesis is organised as follows. Chapter 2 covers the basic principles of irrigation and reviews existing modeling and control approaches. Chapter 3 explains the design of the proposed framework, including the combination of Zone Model Predictive Control, online parameter estimation with Recursive Least Squares (RLS), and real-time estimation of Readily Available Water (RAW). Chapter 4 describes how the simulations were set up in AquaCrop and presents results for different scenarios. Chapter 5 reflects on these results, discussing the main trade-offs, limitations, and their implications in practice. Finally, Chapter 6 summarises the main findings and outlines possible directions for future work.

2

Background

2.1. Fundamentals of Irrigation

In order to irrigate effectively, it is necessary to understand the dynamics of the water in relation to the soil, crop and atmosphere. The soil can be seen as a reservoir where all the water is stored; however, not all the water in the soil is available to the crop. The main goal for an irrigation controller is to manage the soil moisture content in such a way that it is kept at an appropriate level. What is deemed appropriate is dependent on the objective of the controller, this can for example be to avoid water waste, or to maximize crop yield.

Often times, when considering the state of the soil, we are not interested in the entire soil profile, but only in the section that we call the **root zone**. This zone is the depth of the soil where the majority of a crop's roots are concentrated and from which they extract water and nutrients. Focusing on this region, rather than the entire soil profile, allows us to determine the irrigation needs of a crop.

Figure 2.1 shows a diagram that helps visualise the inflows and outflows of moisture in that root zone. Water entering the soil can come from three different sources: **Irrigation**, **Precipitation** (rainfall), and **Capillary Rise**. The latter is the upward movement of water through the soil from deeper, more moist layers towards drier upper layers. If too much water enters the soil, then it can be in a state of **Saturation**, meaning that all the pores in the soil are filled with water. When this happens, excess water can no longer infiltrate the soil. It may leave the system through **Runoff**, which is the movement of water over the soil surface, or through **Deep Percolation**, which is the downward movement of water in the soil, where it goes beyond the root zone and therefore becomes unavailable to the crop.

When discussing the 'moisture level', we are typically referring to one of the following two measurements. The first is volumetric water content (θ), which is a ratio that indicates the volume of water per unit volume of soil (mm^3/mm^3) and is often what sensors measure directly. The second method of measurement is water depth (mm), which represents the total amount of water held within a specific column of soil, often the root zone. This unit is practical because it directly relates the amount of water in the soil to the amount of water added via precipitation or irrigation, both of which we measure in millimetres. The conversion between the two depends on the **Rooting Depth (Z_r)** in meters using

$$Water\ Depth = 1000 \times Z_r \times \theta \tag{2.1}$$

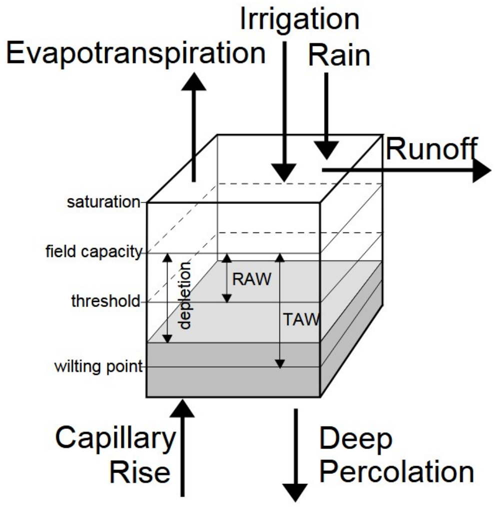

Figure 2.1: Water Soil Dynamics Within the Root Zone [1]

Using this metric, we can define two thresholds that describe the soil's water-holding capacity, which are intrinsic properties of the soil type. **Field Capacity (FC or θ_{FC})** is the optimal upper limit for soil moisture, at which point water and air are readily available to the roots of a crop [1]. After saturation, this is the amount of soil moisture remaining after all the excess has drained away. The **Permanent Wilting Point (WP or θ_{WP})** indicates the opposite, it is the lower limit of soil moisture. When this threshold is reached, the crop can no longer extract water from the soil, and it will start to wilt, resulting in irreversible damage to the crop. The total volume of water that the soil can hold between these two thresholds is referred to as the **Total Available Water (TAW)**. It represents the entire water reservoir accessible to the crop, expressed in mm of water

$$TAW = (\theta_{FC} - \theta_{WP}) \times Z_r \times 1000 \tag{2.2}$$

where θ_{FC} is the water content at field capacity (mm^3/mm^3), θ_{WP} the water content at the wilting point (mm^3/mm^3) and Z_r the rooting depth (m). However, not all water in the TAW is easily accessible. As the soil dries, it becomes harder for the crop to extract water. The **Readily Available Water (RAW)** is the portion of the TAW that a crop can extract without experiencing water stress, which in turn could negatively impact growth and yield. It is defined as

$$RAW = p \times TAW \tag{2.3}$$

where the factor p is a value between 0 and 1, that is dependent on the crop's sensitivity. A value around 0.5 is common.

To determine the irrigation need, the term **Root Zone Depletion** is used in this thesis. It represents the amount of water (in mm) needed to refill the root zone from it's current moisture content, back to Field Capacity [1]. It is determined by

$$RZD = (\theta_{FC} - \theta_i) \times Z_r \times 1000 \tag{2.4}$$

where θ_{FC} is the water content at field capacity (mm^3/mm^3), θ_i the average soil water content for the effective root zone (mm^3/mm^3) and Z_r the rooting depth of the crop(m). Intuitively, this means that when the RZD is at 0 mm, the soil is at Field Capacity and no irrigation is needed. If, for example, the RZD is at $80mm$ and a rainfall event occurs, causing $15mm$ of water to enter the soil, then it is logical to assume that the RZD will now be closer to $65mm$. It is best to think of RZD as a deficit with respect to Field Capacity. Using such a metric is preferred over volumetric moisture content because it accounts for the crop's changing water demand by including the rooting depth (Z_r). As a crop grows, its rooting system expands, increasing the size of TAW, and the RZD value will scale with it correctly.

2.2. Dynamic System Modelling

There are multiple ways of modelling the dynamics of soil moisture in the ground. One way to do so is by using Richard's equation. This is a sophisticated, physics-based description of water flow in the soil [6]. If 'flux' describes the movement of water, then Richard's equation can solve for the flux at every point and time within the soil profile. It can be combined with a 'sink' term that models the water uptake of a crop. When all the relevant parameters are available, the equations can model the soil moisture dynamics accurately. However, its highly non-linear structure makes it hard to use for direct control. An alternative, simple yet effective way of modelling the dynamics of the soil moisture in the root zone is a water balance model.

2.2.1. Water Balance Model

This model tracks the change in soil moisture by summing the inflows and outflows of water into the soil. The inflows are irrigation and precipitation (rainfall), and the outflows are crop evapotranspiration, surface run-off and deep percolation. This relationship can be formed into a linear discrete-time model that predicts the RZD

$$D(k) = c_1 D(k-1) + c_2 ET_c(k) + c_3 P(k) + c_4 I(k) + c_5 S(k) \tag{2.5}$$

where $D(k)$ is the RZD at the end of day k, ET_c is the total crop evapotranspiration for the day k, $P(k)$ is the effective rainfall, $I(k)$ the irrigation depth, $S(k)$ is the saturation amount (includes deep percolation and surface run-off). The coefficients c_1, c_2, c_3, c_4, c_5 represent the system's physical parameters, which can be determined through system identification. The coefficients can be different depending on the crop type, soil, climate, etc. The linear structure of Equation 2.5 makes it a powerful yet simple approximator of RZD, and because of this, it is used in a lot of irrigation control research [9, 10, 16, 4].

2.2.2. Crop Evapotranspiration

A main component of the water-crop dynamics is evapotranspiration. It is the combined loss of water from crop transpiration and soil evaporation. A distinction can be made between Reference Evapotranspiration (ET_o) and Crop Evapotranspiration (ET_c). The first represents the atmospheric demand for water, whereas the latter represents the actual evapotranspiration that a crop undergoes.

Reference Evapotranspiration (ETo)

ET_o is a combination of multiple factors like: solar radiation, air temperature, humidity, wind-speed [25]. It was introduced to create a standardised measure of the atmosphere's evaporative demand. It is defined as the ET rate from a uniform growing vegetation surface (like grass) that is not short of soil water. A popular method for doing so is the Penman-Monteith Method [25], is estimates the ET_o in mm/day by making use of measurements of the solar radiation (sunshine), air temperature, humidity and wind speed.

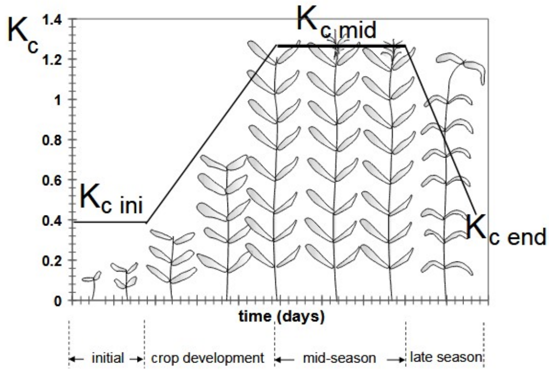

Figure 2.2: Example of a Crop Coefficient Curve [1]

Crop Evapotranspiration (ETc)
ET_c is the water loss from a specific crop under ideal conditions; it differs from ET_o because of the crop's unique characteristics, like height, albedo (reflectance), canopy resistance, etc [1]. One method to calculate this is to calculate it directly using the Penman-Monteith equation, by adjusting parameters like the canopy surface resistance to match the specific characteristics of the crop that is being studied. However, in practice, this is rarely used since the required parameters are difficult to estimate and they change continuously over a crop's lifetime.

The crop coefficient approach is the most common and practical method for calculating ET_c. This method uses the earlier introduced ET_o for the atmospheric demand, and then modifies it with a crop-specific coefficient (K_c). The relationship between these different variables is

$$ET_c = K_c ET_o \tag{2.6}$$

where K_c is the ratio that incorporates the difference between the specific crop and the reference grass surface. The single crop coefficient combines the effect of crop transpiration and evaporation into a single coefficient. This approach is most suited for irrigation planning over longer periods. The dual crop coefficient approach is for more precise day-to-day calculations, as it separates K_c into

$$K_c = K_{cb} + K_e \tag{2.7}$$

where K_{cb} is the basal crop coefficient, which primarily reflects crop transpiration. And K_e is the soil evaporation coefficient, which describes the evaporation coefficient from the soil surface. This dual approach is harder to estimate.

For both approaches, the coefficient K_c changes throughout the season. This trajectory can be represented by a crop coefficient curve, which indicates the coefficient value at different growth stages. Figure 2.2 shows an example of a possible trajectory. To estimate the crop coefficient values over a growing season, lookup tables can be used for a specific crop, climate, and irrigation conditions. Collected ET_c data can be of good help in verifying the coefficients.

2.3. Existing Irrigation Control

Most existing recent research into irrigation control falls into two categories: model-based approaches and data-driven methods. Model-based strategies, particularly Model Predictive Control (MPC), use a mathematical representation of the soil-water-crop system to anticipate future states and calculate the best irrigation actions. On the other hand, data-driven methods like Reinforcement Learning try to learn these patterns directly from historical data without needing an explicit physical model. A distinction worth making is the time horizon being considered: daily irrigation and seasonal planning.

Daily irrigation. Existing literature applies MPC on a daily basis to track a desired soil moisture reference, using weather forecasts over a short prediction horizon to anticipate rainfall and reduce unnecessary irrigation. In [9], the authors develop a Robust MPC using an identified water balance model and explore methods for handling uncertainty in weather forecasts. They find that when forecasts are reasonably accurate, treating them as deterministic is a practical and acceptable approach. This conclusion also explains some of the assumptions made in this thesis. The authors of [19] explore the concept of Zone MPC in an irrigation context. However, in a quite different setting, they apply it to grass on an hourly timescale using a Linear Parameter Varying (LPV) model rather than a daily water-balance model for a field crop. Their results still provide a useful reference point, showing that a zone-based formulation uses less water than a traditional set-point controller.

Seasonal planning A natural extension is to combine a seasonal optimiser with a daily controller, as illustrated in Figure 2.3. The idea is that a high-level planner determines a reference trajectory for the full growing season, accounting for constraints such as limited water supply or logistical restrictions. At the same time, a daily MPC controller tracks that reference. In [16], researchers apply this 2-level structure to sugarcane irrigation in Mozambique, the same climate/crop considered in this thesis, and demonstrate a 30% improvement in water productivity compared to traditional irrigation. Their framework relies on crop-specific information, including the crop's sensitivity to water stress at different growth stages and historical datasets to estimate evapotranspiration from canopy cover. [8] takes a similar 2-level approach but shifts the focus toward economic optimisation, incorporating dynamic water and electricity prices into the seasonal planner. Based on a case study in Spain, they report savings of around 22.8% in water consumption and 43.5% in electricity costs compared to traditional irrigation. Their work highlights the value of treating the soil itself as a water buffer, a concept that is also central to the Zone MPC formulation used in this thesis.

The gap this thesis addresses. Across both categories, a recurring theme is the dependence on prior knowledge. Model-based approaches like those in [9] and [19] require an initial system identification phase using a previously collected dataset, and the resulting model is then kept static throughout the season. Seasonal planners like those in [16] and [8] go further, relying on detailed crop-specific parameters, such as sensitivity to water stress at different growth stages or a pre-computed crop coefficient curve, that are not readily available in many settings. Data-driven approaches face an even greater challenge in this regard, as they typically require far larger historical datasets for training. Across all these methods, the assumption is that the necessary data and prior knowledge are available before the growing season begins, which is often not the case in data-scarce environments or when conditions change significantly between seasons. The framework proposed in this thesis addresses this research gap. It does not rely on historical datasets or crop-specific models, and instead learns the system dynamics online during the growing season using a limited number of measurements. The objective is to investigate whether effective predictive irrigation control is still achievable under these constraints.

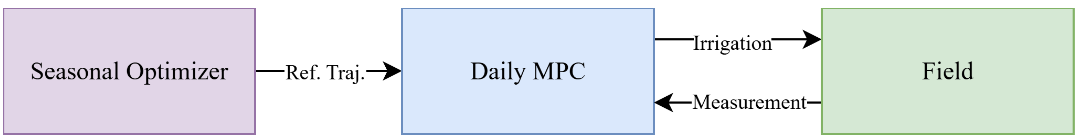

Figure 2.3: 2-level Irrigation Control

3 Design of the Control Framework

In this chapter we design an irrigation framework that can fill the earlier identified research gap. First, the problem is defined, and a general overview of the framework's proposed architecture is presented, including how the components interact. Afterwards each components is discussed in detail and the choices behind them are explained.

3.1. Problem Setup and Assumptions

Before describing the framework design, it is useful to clearly state the assumptions and boundary conditions under which the proposed framework will operate. These define the scope of the problem and will justify several design choices made in subsequent sections.

No prior data or crop-specific knowledge is available. The core design constraint of this framework is that it must be deployable at the start of a growing season without historical datasets, pre-identified models, or crop-specific parameters. The only crop-related value assumed to be available is the depletion fraction p, which determines the RAW threshold. In this thesis, it is treated as a predefined management parameter, as it defines the threshold of acceptable water stress; values for it can also be found in standardised tables.

Scope is limited to daily irrigation control. This framework focuses exclusively on the daily irrigation layer of the control hierarchy described in section 2.3. The seasonal planning problem, which can involve allocating a limited water supply across the season, optimising across multiple fields, incorporating economic constraints such as water and electricity pricing, or making deliberate trade-offs between water use and crop yield, is out of scope. Note that the daily controller can penalise irrigation through its cost function, but this reflects a preference for water efficiency at each time step rather than a long-term seasonal budget or an economic objective. The findings nonetheless remain relevant for seasonal planning research, since any 2-level framework still requires a functioning daily controller.

Weather forecasts are available and treated as accurate. The Zone MPC relies on a forecast of evapotranspiration and precipitation over the prediction horizon. In this framework, forecast data from a local weather station is assumed to be available and is treated as deterministic. In practice, forecasts always carry uncertainty, particularly for the timing and magnitude of rainfall events. This is a common assumption in predictive irrigation research, and its implications are discussed further in section 5.2.

θ_{FC} and θ_{WP} are known. The framework requires knowledge of the soil's water-holding capacity, specifically the volumetric water content at field capacity (θ_{FC}) and the permanent wilting point (θ_{WP}). These are properties of the soil type and can be obtained either through a soil survey or by identifying the soil type and looking up the values. They are not considered part of the system dynamics to be learned and are therefore treated as known constants throughout the framework.

Daily RZD measurements can be obtained. The framework assumes that volumetric soil moisture measurements can be measured, allowing the Root Zone Depletion to be calculated using Equation Equation 2.4. These measurements do not need to be taken every day; a sparse measurement strategy is explored in section 3.6

3.2. Architecture Overview

This section provides a general overview of the main components introduced in the proposed framework and explains how they interact. The overall goal is to construct a general framework for predictive irrigation that does not rely on historical data or prior information, and that can adapt to changing conditions over time. The framework is divided into the following components:

- The **Zone MPC** is responsible for formulating an irrigation strategy that an operator can execute. This control input sequence accounts for the weather forecast over the coming days, allowing the controller to anticipate weather conditions and irrigate optimally. The model used by the Zone MPC to describe the RZD dynamics is identified online during the growing season and is continuously updated using information from the RLS filter. The controller is also provided with a reference to track throughout the season, namely the RAW, which is supplied by the RAW estimator.
- The **Recursive Least Squares (RLS) filter** is responsible for estimating the coefficients of the water balance model during the growing season. On days when measurements are available, we compare the predicted RZD to the measured RZD and update the model coefficients accordingly. At the start of the growing season, the model uses initial coefficient estimates and refines them over time. This approach also allows the model to adapt to changing soil and crop dynamics throughout the season, including variations in ET's effect on the crop. Because the framework assumes no access to historical data, ET_c cannot be modelled beforehand, making real-time estimation necessary.
- The **RAW estimator** is responsible for estimating the Readily Available Water throughout the growing season. The RAW serves as the controller's reference, indicating the soil moisture level at which the crop begins to experience water stress. By keeping the RZD above this threshold, the controller can minimise water usage while also limiting crop water stress. Since the RAW depends on the crop's rooting depth (Z_r), and the evolution of Z_r over the growing season is unknown, the RAW estimator aims to predict this trajectory using a limited number of Z_r measurements. The Zone MPC can then use this estimated trajectory.

Figure 3.1 provides a graphical overview of the different components and how they interact. At the end of each day, the current RZD and the applied irrigation are measured, and the weather conditions for that day are known. The RLS filter uses these inputs, together with the current model coefficients, to compute a predicted RZD. This prediction is then compared with the measured RZD, after which the model coefficients are updated accordingly. The Zone MPC then uses this updated model, the measured RZD from the previous day, and the weather forecast over the prediction horizon to compute an optimal irrigation strategy. The first day of this schedule is applied to the field, and the process is repeated the next day when new measurements become available. The RAW estimator is updated on days when the rooting depth is measured and provides the reference signal for the Zone MPC.

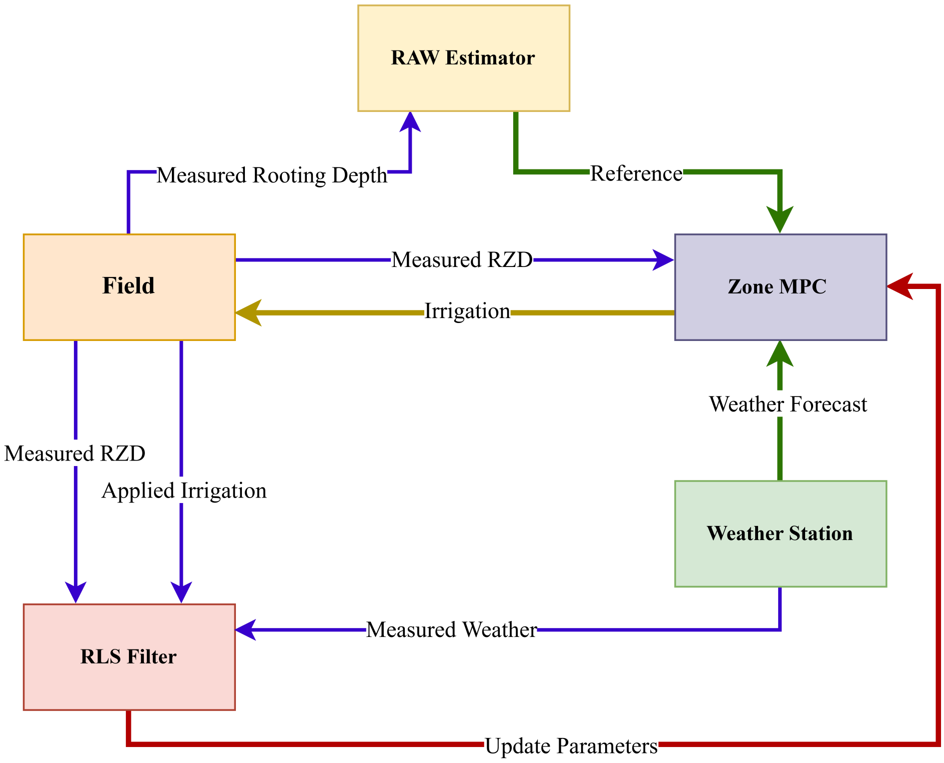

Figure 3.1: Overview of the Irrigation Control Framework

3.3. Zone Model Predictive Control

MPC is a proven and established advanced control strategy that has been widely applied across different industries, and more recently in agricultural systems as well. Unlike classical control methods (such as PID), which primarily react to past errors, MPC is proactive and looks ahead. It uses a model of the system dynamics to predict how the system states will evolve over a chosen prediction horizon.

The core idea behind MPC is the Receding Horizon principle. At every sampling step, the controller solves a finite-horizon optimisation problem. It computes the sequence of control inputs that minimises a given cost function over the prediction horizon. After the optimisation is solved, only the first control input from that sequence is applied to the system. At the next time step, the optimisation problem is solved again. This iterative process enables the controller to correct for unexpected disturbances and modelling errors in real-time. An advantage of MPC is that it can handle complex interactions and enforce constraints on both inputs and states. However, achieving good performance requires an accurate model.

The structure of MPC is particularly well-suited to the challenges in irrigation control. Irrigation is a heavily influenced by external inputs from the weather: soil moisture is depleted by evapotranspiration and replenished by precipitation. Because reasonably accurate weather forecasts are available, MPC can incorporate these inputs. These forecasts allow the controller, for example, to withhold irrigation when rain is expected in the near future, helping to prevent unnecessary water use. Physical limits also constrain agricultural systems; the controller must respect a water balance model and enforce limits on irrigation amounts. Finally, the goal is not only to keep soil moisture within a desirable range but also to balance it against the cost of water.

3.3.1. Set-point vs Zone MPC

Having introduced the general idea behind MPC and how it addresses the challenges of irrigation control, we will briefly consider the choice for Zone MPC in this section.

Most MPC applications use set-point control, where the goal is to drive the system toward a specific desired value. However, this thesis instead utilises Zone MPC. In Zone MPC, the controller aims to keep the system state within an acceptable range rather than at a single precise value. This perspective fits irrigation very naturally. Soil moisture does not have a sharply defined 'correct' value; there is some room for fluctuations. Optimal plant growth does not depend on hitting a single moisture level but on maintaining moisture within a healthy range. The soil–root system is also not highly sensitive or chaotic, meaning minor deviations do not lead to significant or sudden changes in behaviour. For these reasons, it intuitively makes sense to go in this direction.

Although Zone MPC has not been widely applied to irrigation, it has been explored in a related context by [19]. Their results are promising: the zone-based controller outperformed a traditional set-point controller, mainly by using substantially less water. However, the authors pointed out that the performance of the set-point controller depends on the weather conditions and would perform differently in the absence of rain.

3.3.2. Water Balance Model Identification

As previously mentioned, MPC requires a model of the dynamics to function. The model used in this thesis to determine the RZD on a day-to-day basis is a water balance model, as described in subsection 2.2.1. Before introducing the online learning component to the framework, a static model is first identified using historical data. This static model is used as a reference throughout this thesis: it represents what is achievable when prior data and crop-specific knowledge are available. It provides a meaningful point of comparison for the proposed framework.

To estimate the coefficients of Equation 3.1, we use a grey-box system identification approach: the model structure is known, but the coefficients $(c_1, c_2, c_3, c_4, c_5)$ still need to be learned from data.

$$D(k) = c_1 D(k-1) + c_2 ET_c(k) + c_3 P(k) + c_4 I(k) + c_5 S(k) \tag{3.1}$$

where $D(k)$ is the Root Zone Depletion (RZD) at the end of day k, as introduced in subsection 2.2.1. We collect a dataset of length L containing, at each time step k, the RZD, applied irrigation, precipitation, crop evapotranspiration (ET_c), and saturation. The data can be arranged in the linear regression form

$$\underbrace{\begin{bmatrix} D(2) \\ D(3) \\ \vdots \\ D(N) \end{bmatrix}}_{y \in \mathbb{R}^{(L-1)\times 1}} = \underbrace{\begin{bmatrix} D(1) & ET_c(2) & P(2) & I(2) & S(2) \\ D(2) & ET_c(3) & P(3) & I(3) & S(3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D(L-1) & ET_c(L) & P(L) & I(L) & S(L) \end{bmatrix}}_{X \in \mathbb{R}^{(L-1)\times 5}} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}}_{\theta \in \mathbb{R}^{5\times 1}} \tag{3.2}$$

Because X contains more rows than columns ($(L-1) \gg 5$), the system is overdetermined and the parameters can be estimated using the least-squares solution, which minimizes the sum of squared differences between the measured data and the model output. This yields estimates for the coefficients c_1, c_2, c_3, c_4, and c_5 in Equation 3.1.

The linear model for RZD (Equation 3.1), can then be defined as a state space system

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k) \tag{3.3}$$

where the state $x(k)$ is $D(K)$, and the control input $u(k)$ is the irrigation $I(k)$. The precipitation and crop evapotranspiration are considered as disturbances, which are defined as

$$d(k) = \begin{bmatrix} ET_c(k) \\ P(k) \end{bmatrix} \tag{3.4}$$

with $A = c_1$, $B = -c_4$, $B_d = \begin{bmatrix} c_2 \\ -c_3 \end{bmatrix}^T$.

Because the saturation level $S(k)$ cannot be measured or used in real-time irrigation control, both this term and the coefficient c_5 are excluded from the model. This simplification is justified, as an optimally performing controller should prevent over-irrigation [10]. Nonetheless, since saturation may occur during the training phase, the coefficient c_5 is retained in the system identification process.

3.3.3. Zone MPC - Mathematical Formulation

Since we now have the model and justified why Zone MPC is chosen. A mathematical formulation is given for it's structure. The cost function over a prediction horizon of size N is defined as

$$J = \sum_{k=0}^{N-1} (s_1(k)^2 Q_{upper} + s_2(k)^2 Q_{lower} + Ru(k)^2) \tag{3.5}$$

with $Q_{upper}, Q_{lower} \geq 0$, which determine the penalty for exceeding the zone bounds on the top and bottom of the target zone, respectively. And the input $R > 0$, which penalises the magnitude of the input. The variables $s_1(k)$ and $s_2(k)$ are slack variables. The constraint for the linear dynamics is the following

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k) \tag{3.6}$$

The constraint for the initial state is

$$x(0) = x_{init} \tag{3.7}$$

and the constraints that outline the target zone are

$$\begin{aligned} x_{ref}(k) + l_{upper} + s_1(k) &\geq x(k) \\ x_{ref}(k) - l_{lower} + s_2(k) &\leq x(k) \\ s_1(k), s_2(k) &\geq 0 \end{aligned} \tag{3.8}$$

Where $l_{upper}, l_{lower} \geq 0$ determine the size of the target zone, relative to the given reference $x_{ref}(k)$. See Equation 3.5. The constraints for the state and input bounds are the following:

$$\begin{aligned} x_{min} \leq x(k) \leq x_{max} \\ u_{min} \leq u(k) \leq u_{max} \end{aligned} \tag{3.9}$$

Which results in the Zone MPC Problem:

$$\begin{aligned}
\min_{u(k),x(k),s_1(k),s_2(k)} \quad & \sum_{k=0}^{N-1}(s_1(k)^2 Q_{upper} + s_2(k)^2 Q_{lower} + Ru(k)^2) \\
\text{subject to} \quad & x(k+1) = Ax(k) + Bu(k) + B_d d(k), k = 0, \ldots, N-1 \\
& x(0) = x_{init} \\
& x_{min} \leq x(k) \leq x_{max}, k = 0, \ldots, N-1 \\
& u_{min} \leq u(k) \leq u_{max}, k = 0, \ldots, N-1 \\
& x_{ref}(k) + l_{upper} + s_1(k) \geq x(k), k = 0, \ldots, N-1 \\
& x_{ref}(k) - l_{lower} + s_2(k) \leq x(k), k = 0, \ldots, N-1 \\
& s_1(k), s_2(k) \geq 0, k = 0, \ldots, N-1
\end{aligned} \tag{3.10}$$

3.4. Online Model Estimation with Recursive Least Squares (RLS)

The goal of the online model estimator is to estimate the coefficients of the water balance model. The MPC controller then uses these coefficients to predict the expected root zone depletion (RZD) over the prediction horizon. Estimating the model online allows the controller to operate without prior knowledge of the system and to adapt to changing conditions over time. If the controller makes use of ET_o then it is fully operating without prior knowledge, it makes use of ET_c then it still presumes to know the changing effect of ET on a crop over the growing season.

To achieve this, we introduce a learning mechanism based on Recursive Least Squares (RLS). RLS is an adaptive filtering method that recursively estimates model coefficients by minimising a weighted least squares cost function [18]. Unlike batch least squares, which uses all available data at once, RLS updates its parameter estimates incrementally as new data points become available. These properties make it well-suited for real-time applications where system dynamics may vary over time.

The water balance model introduced in Equation 3.1 uses crop evapotranspiration (ET_c) as an input. As discussed in subsection Equation 2.6, ET_c is related to the reference evapotranspiration through the crop coefficient: $ET_c = K_c ET_o$ Incorporating this into Equation 3.1 yields:

$$D(k) = c_1 D(k-1) + c_2 K_c(k) ET_o(k) + c_3 P(k) + c_4 I(k) \tag{3.11}$$

When ET_c is used/known, the term $c_2 K_c(k)$ becomes a single coefficient, since K_c is known. However, when ET_c is not available, which is the case when we don't have access to prior information, then ET_o from a weather station can be used instead. In that case we define a combined coefficient $\bar{c_2} = c_2 K_c(k)$, which absorbs the the seasonal variation in K_c that the framework can no longer model explicitly. The model then becomes

$$D(k) = c_1 D(k-1) + \bar{c_2} ET_o(k) + c_3 P(k) + c_4 I(k) \tag{3.12}$$

Because K_c varies over the growing season, $\bar{c_2}$ is inherently time-varying. The goal of the RLS estimator is therefore to track this evolving coefficient online, along with other model parameters. The distinction between using ET_c and ET_o as an input is important, and its implications for the estimators performance are evaluated in section 4.5

3.4.1. Theoretical Formulation

To achieve this, the RLS algorithm recursively estimates the parameter vector $\theta(k)$ of a linear model at time step k. In this thesis, these parameters correspond to the coefficients of the water balance model introduced in Equation 3.12. The model predicts the output $D(k)$ using an input vector $\phi(k-1)$ that contains past measurements. When ET_c is available, $\phi(k-1) = [D(k-1), ET_o(k), P(k), I(k)]^T$ and the parameter vector $\theta = [c_1, c_2, c_3, c_4]^T$. When only ET_o is available, ET_c is replaced by ET_o in the regressor, and the corresponding coefficient $\bar{c_2}$ now represents the combined effect of c_2 and the current K_c.

The model relationship is given by

$$\hat{D}(k) = \phi(k-1)^T \theta(k-1) \tag{3.13}$$

At each time step k, once a new measurement $D(k)$ becomes available, the RLS algorithm updates the parameter vector from its previous value $\theta(k-1)$ to a new estimate $\theta(k)$. This update consists of the following steps [18]:

1. **Prediction and Error Calculation:** First, the output is predicted using the parameter estimates from the previous time step. The prediction error $\varepsilon(k)$ is computed as the difference between the measured output and the predicted output.

$$e(k) = D(k) - \phi(k-1)^T \theta(k-1) \tag{3.14}$$

2. **Gain Vector Calculation:** Next, the gain vector $g(k)$ is calculated. This vector determines the sensitivity of the parameter update to the current prediction error

$$g(k) = \frac{P(k-1)\phi(k-1)}{\lambda + \phi(k-1)^T P(k-1)\phi(k-1)} \tag{3.15}$$

where $P(k-1)$ is the covariance matrix representing the uncertainty of the estimates, and λ is the forgetting factor ($0 < \lambda \leq 1$). Choosing $\lambda < 1$ reduces the influence of older data, allowing the algorithm to adapt more quickly to changes in the system dynamics.

3. **Parameter Update:** The parameter vector is updated by adding a correction term proportional to the gain vector and the prediction error

$$\theta(k) = \theta(k-1) + g(k)e(k) \tag{3.16}$$

4. **Covariance Matrix Update:** Finally, the covariance matrix is updated to reflect the information gained from the new data point. This prepares the algorithm for the next iteration.

$$P(k) = \frac{1}{\lambda}\left(P(k-1) - g(k)\phi(k-1)^T P(k-1)\right) \tag{3.17}$$

This recursive procedure allows the RLS algorithm to gradually converge to suitable model parameters while continuously adapting them as the system changes.

3.4.2. Parameter Tuning and Initialisation

The performance of the RLS estimator strongly depends on how three parameters are initialised and tuned: the forgetting factor λ, the initial covariance matrix $P(0)$, and the initial parameter vector $\theta(0)$.

Forgetting Factor (λ)

The forgetting factor λ controls how much past data influences the parameter estimates. It allows the estimator to discard older data in favour of recent measurements, enabling the tracking of time-varying parameters. When $\lambda = 1$, the algorithm reduces to the standard recursive least squares method with infinite memory, meaning all data points are weighted equally. This results in strong noise suppression (low variance), but it also prevents the estimator from adapting to changes in the system dynamics. For systems with slowly varying dynamics, λ is typically chosen between 0.95 and 0.99. Lower values of λ allow for faster tracking of parameter changes, but at the cost of increased sensitivity to noise [18].

Initial Parameter Vector ($\theta(0)$)

The vector $\theta(0)$ provides the initial guess for the parameter estimation. In this case, the elements of $\theta(0)$ are initialised with a value between -1 and 1.

Initial Covariance Matrix ($P(0)$)

The matrix $P(k)$ reflects the confidence in the current parameter estimates. The initial covariance $P(0)$ is commonly set as a diagonal matrix

$$P(0) = \alpha I \tag{3.18}$$

where I is the identity matrix and α is a positive scalar [3]. A large value indicates high uncertainty in the initial parameter estimates $\theta(0)$. As a result, the algorithm responds strongly to the first measurements, leading to fast initial convergence. In contrast, a smaller value of α implies greater confidence in the initial parameters and therefore results in more conservative updates. In this case, since we assume we know very little about the system, our initial guess will probably not be accurate. Therefore, a relatively large α is then preferred to achieve fast convergence from the initial guess.

3.4.3. RLS and Zone MPC Integration

The RLS algorithm is embedded directly in the control loop, enabling online adaptation of the model. At each time step k for which a measurement of the RZD ($D(k)$) is available, the following steps are performed:

1. Construct regressor vector $\phi(k-1)$ using system inputs and state from previous time step
2. Supply the newly measured RZD, $D(k)$, as the target output
3. Compute the updated parameter vector $\theta(k)$ using the RLS algorithm
4. Update the linear model parameters (A, B, B_d) using the estimated coefficients from $\theta(k)$
5. Solve the MPC optimisation problem for the next time step

This results in a control scheme in which the internal model of the MPC is updated in real time. As a consequence, the controller can learn the appropriate configuration of the water balance model for the given crop–soil combination and potentially adapt to changes over the growing season. It is important to note that this update process relies on the availability of new measurements.

3.4.4. Parameter Constraints and Physical Bounds

At the start of a growing season, new measurements can have a relatively large influence on the update of θ. This can temporarily cause some coefficients to take on unrealistic values. For example, the irrigation coefficient might suddenly become negative, implying that irrigation removes moisture from the soil, something physically impossible. Since the updated parameters are used directly in the MPC loop, unrealistic values can cause the controller to behave poorly and potentially apply excessive irrigation, which is undesirable. While the RLS filter would likely correct itself once more measurements become available, the intermediate use of these parameter updates to compute irrigation makes it better to avoid such situations altogether. For this reason, we would like to impose bounds on specific model coefficients. These bounds are enforced during the parameter update step of the RLS + MPC algorithm. If a coefficient exceeds its predefined bounds, it is clipped to the corresponding bound value.

3.5. RAW Estimator

The final component of the control framework is the Readily Available Water (RAW) estimator. As discussed in section 3.3, the Zone MPC requires a reference trajectory for the desired RZD over the growing season, which is then used to define the boundaries of the target zone. In this framework, the RAW is chosen as the reference, since it represents the threshold at which the crop begins to experience water stress. The closer one stays to that threshold, the more water the controller can save.

The main challenge in determining the RAW is that it depends on the crop's rooting depth (Z_r), which changes over time. Because this framework assumes no prior knowledge of the specific crop growth curve or historical data, the trajectory of Z_r must be estimated online using measurements. Ideally, as few measurements as possible are used, since measuring rooting depth is relatively challenging compared to measuring volumetric water content.

As shown in [20], rooting depth across different crop types generally develops similarly, following the shape of a sigmoid logistic function. Most root growth occurs early in the season, after which the roots become fully developed and no longer grow. This underlying structure can be exploited when designing an estimator for the rooting depth. An initial idea was to fit a logistic function to the incoming data points and use the extrapolation of this fit as a best estimate of the rooting depth over the MPC prediction horizon. However, this approach proved to be too unstable in practice. As a result, a simpler piecewise linear approximation was chosen instead.

On days when a manual rooting depth measurement is available, the estimator updates its internal growth model. Root growth is modelled as a linear relationship between consecutive measurement intervals. The growth rate G_r is computed as

$$G_r = \frac{Z_r(k_{meas}) - Z_r(k_{prev})}{k_{meas} - k_{prev}} \tag{3.19}$$

where k_{meas} denotes the index of the current measurement day and k_{prev} the index of the previous measurement day. This growth rate is then used to extrapolate the future rooting depth over the MPC prediction horizon. For any future day i relative to the current day k, the predicted rooting depth $\hat{Z}_r(k+i)$ is given by

$$\hat{Z}_r(k+i) = Z_r(k_{meas}) + i \cdot G_r \tag{3.20}$$

The estimator also includes a mechanism to detect growth stagnation. If the change in measured rooting depth falls below a predefined threshold, the estimator assumes that the crop has reached its maximum rooting depth for the season, and Z_r is held constant from that point onward. Additionally, if the crop's maximum rooting depth is known, the estimator can prevent overshoot of $\hat{Z}_r$. In that case, the predicted rooting depth is computed as

$$\hat{Z}_r(k+i) = \min\big(Z_r(k_{meas}) + i \cdot G_r, Z_{max}\big) \tag{3.21}$$

where Z_{max} represents a predefined physical limit on the rooting depth.

Once the rooting depth has been estimated, the RAW can be computed using Equation 2.3. The fraction p is assumed to be known, as learning it online is inherently tricky. Doing so would require the controller to deliberately allow the crop to enter water stress in order to identify the "tipping point," which is counterproductive for an irrigation controller designed to avoid stress. In addition, the FAO provides standard tabulated values for p for a wide range of crops, as reported, for example, in [1].

At each time step k, the estimator outputs a vector of RAW values with a length equal to the prediction horizon N. Although the estimator is only updated on days when measurements are available, it generates a daily "best-guess" trajectory based on the most recent estimate of G_r. This process allows the MPC to anticipate the increasing water demand of a growing crop and adjust the irrigation schedule proactively, even when Z_r measurements are sparse.

3.6. Sparse Soil Moisture Measurements

In research where some form of MPC is applied to regulate soil moisture levels, the controller typically receives a new measurement of the actual soil moisture state at each time step, usually once per day. However, many farmers, especially those with smaller operations, do not have access to advanced soil moisture monitoring systems like those found in large greenhouses. Manual measurement methods, such as using a tensiometer, require both labour and time. This creates a compelling reason to investigate how sparse soil moisture measurements influence the effectiveness of moisture regulation. This is a particularly relevant aspect to explore, as the accuracy of the MPC's model becomes more critical when fewer real-world measurements are available. Within the MPC framework, the receding horizon property can partially compensate for inaccuracies in the dynamics. To incorporate sparse measurements into the control problem, we define a mechanism that captures the periodic availability of RZD measurements based on a fixed interval m.

3.6.1. Fixed Interval Measurements

The measurement interval m defines how frequently true moisture measurements are used to correct the prediction. Measurements are taken on days

$$k \in \mathbb{M} = \{m, 2m, 3m, \ldots\} = \{nm : n \in \mathbb{N}\} \tag{3.22}$$

and the binary indicator variable $\delta(k) \in \{0, 1\}$, representing whether a real moisture measurement is available at time step k, is defined as

$$\delta(k) = \begin{cases} 1 & \text{if } k \in \mathbb{M} \quad \text{(measurement)} \\ 0 & \text{otherwise} \quad \text{(model estimation)} \end{cases} \tag{3.23}$$

The state used by the MPC controller, denoted $\hat{x}(k)$, is updated according to

$$\hat{x}(k) = \begin{cases} x(k), & \text{if } \delta(k) = 1 \quad \text{(real measurement)} \\ \hat{x}(k), & \text{if } \delta(k) = 0 \quad \text{(model prediction)} \end{cases} \tag{3.24}$$

where the predicted state is computed as

$$\hat{x}(k) = f(\hat{x}(k-1), u(k-1)) \tag{3.25}$$

and $f()$ represents the model.

3.6.2. Sparse Measurement Strategy - Design

While the fixed-interval approach provides a baseline for studying the effect of sparse soil moisture data, a more refined measurement strategy can be designed. The idea is to reduce the total number of measurements while still maximising the information gained on the days when measurements are taken. Intuitively, this makes sense: if soil moisture is rarely measured on rainy days, it becomes difficult to properly estimate the effect of rainfall on the soil and the crop. In addition, crops generally change more rapidly at the beginning of the season than toward the end, so it is reasonable to measure more frequently early on and gradually reduce the measurement frequency as the season progresses. Based on this intuition, a sparse measurement strategy has been developed. First, so-called "heartbeat" measurements are introduced. These are periodic soil measurements taken every few days, with the interval depending on the stage of the growing season. A starting measurement frequency f_{start} and an ending frequency f_{end} are defined, and the measurement interval gradually shifts from the former to the latter over the course of the season. Second, event-triggered measurements are included, based on irrigation and rainfall events. In some climates, rainfall may be absent for long periods. When rain occurs again, it is important to measure the soil shortly afterwards to capture its effect. The same applies to irrigation: during wet periods, irrigation may not be needed for some time, but when it is applied, measuring its impact helps improve the model accuracy. If more than τ consecutive days pass without a measurement of rainfall for example, an additional measurement is triggered. This acts as a safeguard in case the regular "heartbeat" schedule does not capture these events. For ET_0, such event-based measurements are not necessary, since it changes gradually and remains relatively stable over the season. The measurement strategy is defined more formally below.

The measurement frequency decreases linearly over the season from a starting frequency f_{start} to an ending frequency f_{end} (e.g $f_{start} = \frac{1}{2}$ and $f_{end} = \frac{1}{8}$). The frequency at day k is defined as

$$f(k) = f_{\text{start}} + k \cdot \frac{f_{\text{end}} - f_{\text{start}}}{K} \tag{3.26}$$

where K is the total season length in days. The corresponding measurement interval at day k is then $\Delta(k) = 1/f(k)$. A heartbeat measurement is triggered on day k if the elapsed time since the last heartbeat measurement exceeds the current interval. Defining k_{prev} as the day of the most recent heartbeat measurement, the trigger condition is

$$\delta_{\text{heartbeat}}(k) = \begin{cases} 1 & \text{if } k - k_{\text{prev}} \geq \Delta(k) \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

Upon triggering, k_{prev} is updated to k. This formulation directly captures the intended behaviour: measurements are taken more frequently early in the season and less frequently as the season progresses.

The event-triggered measurements monitor rainfall $P(k)$ and irrigation $I(k)$. Two counters C_{rain} and C_{irr} track the number of days elapsed since the last measurement following their respective events. At each time step, both counters are incremented by one. A measurement is triggered if an event occurs and the corresponding counter exceeds the threshold τ

$$\delta_{\text{event}}(k) = \begin{cases} 1 & \text{if } I(k) > 0 \text{ and } C_{\text{irr}} > \tau \\ 1 & \text{if } P(k) > 0 \text{ and } C_{\text{rain}} > \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

Upon a successful measurement where $\delta(k) = 1$, the counters are reset according to

$$C_{\text{rain}} \leftarrow 0 \quad \text{if } P(k) > 0, \qquad C_{\text{irr}} \leftarrow 0 \quad \text{if } I(k) > 0 \tag{3.29}$$

That is, a counter is reset only if its corresponding event occurred on the measurement day. If a measurement is triggered by one event but not the other, only the relevant counter is reset. The combined measurement indicator is then

$$\delta(k) = \max\left(\delta_{\text{heartbeat}}(k), \delta_{\text{event}}(k)\right) \tag{3.30}$$

and the state update $\hat{x}(k)$ follows the same logic as defined in Equation 3.24 and Equation 3.25.

4

Simulation Results & Evaluation

In this chapter, the proposed framework from the previous chapter is evaluated. First, AquaCrop is introduced as the simulation model used to test the irrigation strategies in this thesis, along with a description of the specific scenario considered. Next, several performance metrics are defined to assess and compare irrigation schedules across simulations. After that, individual components of the irrigation framework are evaluated separately. This helps isolate their behaviour more effectively and understand its specific effects. Finally, the complete irrigation framework is tested as an integrated system.

4.1. AquaCrop Simulator

Since crops require months to reach maturity, relying solely on field experiments to test control algorithms is impractical. It would require growing multiple generations of crops to gather sufficient data on how different control inputs affect final yield. For this thesis, the AquaCrop model, developed by the Food and Agriculture Organisation (FAO) of the United Nations, was selected as the simulation environment. AquaCrop is a water-driven crop simulation model designed to simulate the yield response to water of several crops. It is particularly well-suited for scenarios where water is a limiting factor during a crop's lifetime [13]. The model has been widely validated across diverse climatic regions and crop types, which ensures that the results of the simulations are credible. It can track the state of various components of the crop, such as the roots and the canopy cover. Aquacrop can be given a large set of parameters that describe the soil state, the crop and its properties, and the groundwater for a particular scenario. In combination with local weather data, Aquacrop can simulate a wide range of agricultural scenarios for all types of climate and crops.

4.2. Case Study and Data

Although this thesis aims to develop a general framework applicable to a wide range of crops and climates, a single agricultural scenario was selected to evaluate the proposed irrigation control techniques. This section describes that scenario and the data used throughout all simulations in this chapter.

The case study focuses on open-field irrigated sugarcane in southern Mozambique. The simulation setup largely follows the configuration presented in [16]. Using a scenario introduced by existing research keeps this thesis realistic, without the effort required to configure and validate a crop-soil model in Aquacrop from scratch. This way, the emphasis remains on the proposed control framework and its performance.

Sugarcane is one of the most important crops in Mozambique and plays a significant role in the country's economy; it makes up 20% of the country's agricultural export [11]. At the same time, it is a highly water-intensive crop, making effective irrigation management essential. Improving irrigation efficiency for sugarcane can therefore lead to significant water savings.

The selected region in southern Mozambique has a tropical climate, with a wet season from December to March and a dry season from May to November [21]. Rainfall is often irregular, with short periods of high-intensity precipitation followed by extended dry periods during the growing season. As a result, irrigation is necessary to ensure stable crop growth. Weather data from 2013 to 2018 is used as input to the AquaCrop model, including daily precipitation, temperature, and reference evapotranspiration.

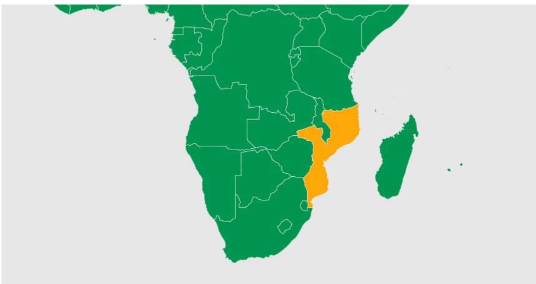

Figure 4.1: Location of Mozambique [5]

Figure 4.2: Sugarcane crop in Mozambique [22]

4.3. Performance Metrics

To analyse the performance of the control strategy, a set of complementary metrics is defined. Together, these metrics evaluate the system from different perspectives. Simulation results are aggregated per growing season, converting daily model outputs into seasonal totals. Three metrics are used to capture the trade-offs between water savings and yield maximisation.

The **Biomass Water Productivity** (w_p) normalises yield by beneficial water use (Transpiration) [23]. It is calculated as

$$w_p = \frac{Y}{\sum T_r} \times 100 \tag{4.1}$$

where Y is the fresh yield in $tonne/ha$ and T_r is transpiration in mm, resulting in units of kg/m^3. This metric reflects how efficiently the crop converts transpired water into biomass. Since transpiration is essential for biomass growth, w_p is expected to remain relatively stable. Significant deviations may indicate periods of severe water stress that negatively impacted yield formation.

In addition, the **Irrigation Water Use Efficiency** ($IWUE$) is introduced [14]. This metric measures the yield (Y) produced per unit of irrigation water applied (I) and is defined as

$$IWUE = \frac{Y}{\sum I} \times 100 \tag{4.2}$$

where Y is the fresh yield in $tonne/ha$ and I is irrigation in mm, again resulting in kg/m^3. This metric is sensitive to the controller's ability to make use of precipitation events. A high $IWUE$ suggests that rainfall was effectively used to support crop growth, with irrigation applied only when necessary to avoid yield-reducing stress. Conversely, a low $IWUE$ indicates over-irrigation or poor use of available precipitation. In relatively wet seasons, this metric is therefore expected to be high.

Finally, the **Water Use Efficiency** (WUE) metric is considered [14]. This metric evaluates the controller's ability to minimise water losses such as runoff and deep percolation. It is defined as the ratio of transpiration to total water inflow

$$WUE = \frac{\sum T_r}{\sum I + P} \tag{4.3}$$

where P is the precipitation in mm, this ratio ranges between 0 and 1 and represents the fraction of total available water that is taken up by the roots and transpired by the crop. A low WUE indicates that a large portion of the supplied water was lost to the environment, suggesting suboptimal irrigation timing by the controller.

4.4. Zone MPC Evaluation

Before introducing online learning, this section evaluates the Zone MPC using a static, pre-identified model and compares it to a set-point MPC. The goal is to establish a baseline performance and confirm that the Zone MPC behaves as expected before adding more complexity. We assume to have access to daily RZD measurements, the evolution of ET_c over the growing season, and a precomputed reference trajectory. The Zone MPC design described in section 3.3 is used.

4.4.1. System identification

To obtain the static model used as a baseline throughout this chapter, a system identification procedure is performed. This subsection describes how the water balance model coefficients are estimated and validated. One season of data was collected ($L = 300$) using a random irrigation strategy, and the steps described in subsection 3.3.2 were followed to identify the system. The weather data came from 2013, and the irrigation strategy assigned each day a 35% chance of an irrigation event, with the amount applied randomly chosen between 0 and 10 millimetres.

The identified coefficients were: $c_1 = 0.991730$, $c_2 = 0.949683$, $c_3 = -0.954503$, $c_4 = -0.816979$ and $c_5 = 1.022286$. The model was then validated on independent test data using the same irrigation strategy, with weather data from 2016, to assess its generalisation capability. Validation was performed using One-Step-Ahead (OSA) prediction and Free-Run Simulation, as shown in Figure 4.3 and Figure 4.4.

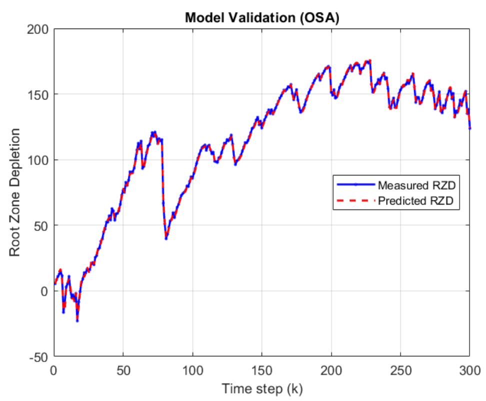

Figure 4.3: Model Validation using OSA Prediction

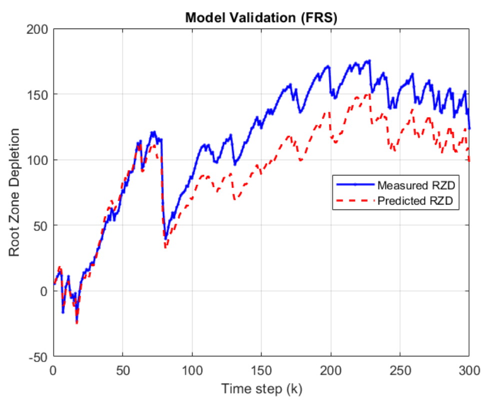

Figure 4.4: Model Validation using Free-Run Simulation

In the OSA configuration, the model receives the measured RZD from the previous time step. This test isolates the model's ability to capture immediate dynamics. The model achieved a high accuracy, with a Root Mean Squared Error (RMSE) of 1.2051, indicating that the identified coefficients effectively capture the daily water balance dynamics. In the Free-Run Simulation, the model predicts RZD recursively over the entire horizon, as it is an auto-regressive process. As seen in Figure 4.4, the model follows the general trajectory of RZD well. However, a deviation occurs around day 75 during a significant precipitation event, where the model overestimated the reduction in depletion, causing a downward shift in the predicted trajectory. This offset persisted for the rest of the validation simulation, resulting in a higher RMSE of 26.14.

Despite this offset in the Free-Run Simulation, the model is suitable for the proposed MPC application. Because MPC operates on a receding horizon principle, the controller is re-initialised with the measured RZD at each time step, effectively resetting any accumulated error.

4.4.2. K_c Modelling

This subsection describes how the crop coefficient K_c is determined for sugarcane in Mozambique, which is required to compute ET_c as input to the static Zone MPC model. In this thesis, ET_c is modelled using the single crop coefficient approach $ET_c = K_c ET_0$ described in subsection 2.2.2. To determine the right values of K_c, the FAO crop evapotranspiration guidelines were used [1]. For sugarcane in a sub-humid climate, the initial recommended coefficients are: $K_c ini = 0.4$, $K_c mid = 1.25$, and $K_c end = 0.75$. These coefficients were then checked in an AQUACROP simulation of a growing season and the values were slightly adjusted to better match the current scenario. The resulting values are: $K_c ini = 0.5$, $K_c mid = 1.1$, and $K_c end = 0.71$. The initial period lasts 45 days, while the mid-season period lasts 285 days. Values of K_c between these three points are calculated using linear interpolation. Figure 4.5 shows the resulting trajectory of K_c over a growing season.

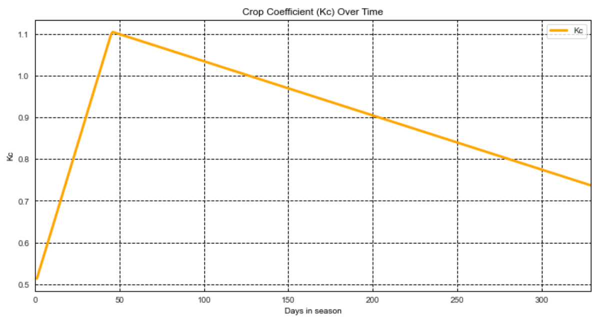

Figure 4.5: Trajectory of K_c over a growing season

4.4.3. Reference Trajectory

The Zone MPC requires a reference trajectory for the desired RZD over the growing season. This subsection describes how that reference is constructed from the RAW for the static baseline controller.

As mentioned in section 3.5, the reference followed by this controller is based on the RAW. Its value changes throughout the season, as it depends on the crop's rooting depth (see Equation 2.2 and Equation 2.3). The rooting depth used in this work is obtained from data collected during a previous growing season, where the crop was sufficiently irrigated, allowing the root system to develop fully. The sensitivity value (p) for sugarcane is taken from an FAO guide on crop water requirements [1]. Together with the other parameters introduced in section 2.1, this allows the RAW to be calculated. The Zone MPC controller uses the RAW, with a slight offset, as the reference to track during the growing season. This offset is introduced to provide a small margin for error. As shown in Figure 4.6, the RAW trajectory curves upward during the first 60 days or so. During this period, the sugarcane root system continues to develop, resulting in an increasing rooting depth. After this phase, the value of the RAW remains constant.

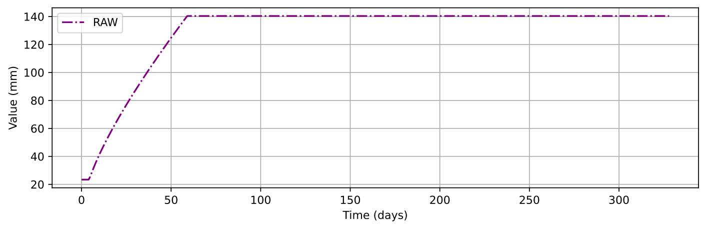

Figure 4.6: Trajectory of RAW over a growing season

4.4.4. Zone MPC - Simulation

A simulation is performed in AQUACROP with the Zone MPC controller as formulated in subsection 3.3.3. The aim is to verify that the controller maintains the RZD within the target zone, makes sensible use of weather forecasts, and avoids unnecessary water use. The growing season that is simulated ranges from 2015/02/10 to 2016/01/05, and makes use of real weather data. The parameters of the controller are tuned by trial and error and can be seen in Table 4.1. ET_c data is given to the controller instead of ET_o, this means that the ET_o data is corrected for the amount of ET that sugarcane undergoes at each point in time during it's lifetime (see subsection 2.2.2).

Table 4.1: Parameters - Zone MPC Simulation

Parameters	Values
Prediction Horizon (N)	10
Input Cost (R)	100
State cost (Q_{upper})	100
State cost (Q_{lower})	1
Zone size (l_{upper})	15
Zone size (l_{lower})	25
Reference offset	20

Figure 4.7 shows three plots that highlight different parts of the irrigation system. The first plot displays the soil state, the second plot shows the irrigation applied by the controller, and the third plot illustrates the weather conditions over the growing season. If we first examine the weather during this season, we can see that precipitation is quite sporadic: it does not occur very often, but when it does, it comes in large amounts. It is also worth noting that the ET_c in mid-season is almost half of its value at the beginning and end of the season.

Looking at the first two plots, we can see that the controller behaves as expected. The RZD mostly stays below the upper edge of the zone. When a significant precipitation event is approaching, the irrigation controller reduces or stops irrigation to anticipate the rainfall and save water. Examples of this behaviour can be seen around days: 70, 210, 250, and 280. In some cases, the controller slightly violates the upper bound of the zone because the upcoming precipitation allows for greater water savings. In these situations, the cost of irrigation outweighs the cost of violating the soft state constraint. It is also noticeable that during the first 50 days of the growing season, irrigation is minimal. During this period, the crop's root system is still developing, allowing it to increase the amount of water it can take up. Precipitation is sufficient for the controller to avoid additional irrigation.

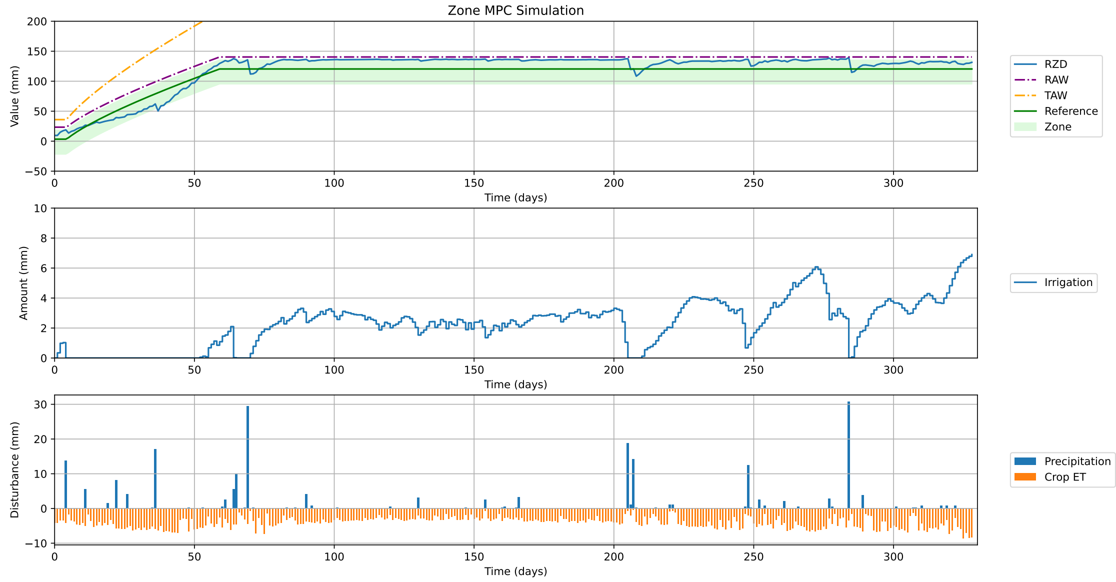

Figure 4.7: Zone MPC simulation (2015-2016)

Table 4.2 shows the values of the performance metrics for the simulated growing season, these metrics are introduced in section 4.3. A simulation is also run over the same growing season using a Field Capacity Setpoint Irrigation (FCSI) controller. This simple controller aims to keep the RZD at 90% Field Capacity. While it is not a sophisticated control strategy, it serves as a useful reference for comparing the performance of the Zone MPC.

It can be seen that the biomass water productivity (w_p) for the Zone MPC is similar to that of the FCSI. Since this value remains stable, it indicates that the Zone MPC controller does not expose the crop to periods of excessively severe water stress. The FCSI controller is a valuable reference in this respect, as it always aims to saturate the field, making water stress very unlikely, albeit at the cost of increased irrigation. The irrigation water use efficiency ($IWUE$) for the Zone MPC is 3 kg/m^3 higher than that of the FCSI controller. This is expected since the Zone MPC uses precipitation more effectively, whereas the FCSI strategy aims to keep the RZD at a fixed level regardless of upcoming rainfall or ET. This can lead to unnecessary irrigation and wasted water, resulting in lower efficiency. Examining the water use efficiency (WUE), we can observe that the proportion of total water taken up by the roots is substantially higher for the Zone MPC compared to the FCSI controller. This shows that the Zone MPC is more effective at minimizing such as run-off and deep percolation.

Irrigation Strategy	$w_p(kg/m^3)$	$IWUE(kg/m^3)$	WUE $(0-1)$
1) Zone MPC	8.492	10.613	0.943
2) Field Capacity Setpoint Irrigation	8.437	7.018	0.706

Table 4.2: Performance Metrics - Zone MPC Simulation (2015-2016)

4.4.5. Zone MPC vs Set-Point MPC

A test was also carried out to highlight the difference in optimisation strategy between the Zone MPC and a set-point MPC. Figure 4.8 shows two simulations run over the same time period but using different control strategies: one with a set-point MPC and one with a Zone MPC. The figure is zoomed in on a precipitation event around day 210 and illustrates how the two controllers respond differently. The set-point controller optimises a quadratic cost function that penalises absolute deviations from the reference, without distinguishing between upward and downward deviations of the RZD. As a result, the controller irrigates to minimise the quadratic deviation on both sides of the reference. In contrast, the Zone MPC does make this distinction, as there is no penalty for the RZD deviating below the reference (within limits). The latter is more suitable for this system, since a downward deviation from the reference corresponds to a lower RZD, which effectively means additional water stored in the soil. The set-point control strategy can therefore lead to unnecessary irrigation and water stress for the crop, which Zone MPC avoids.

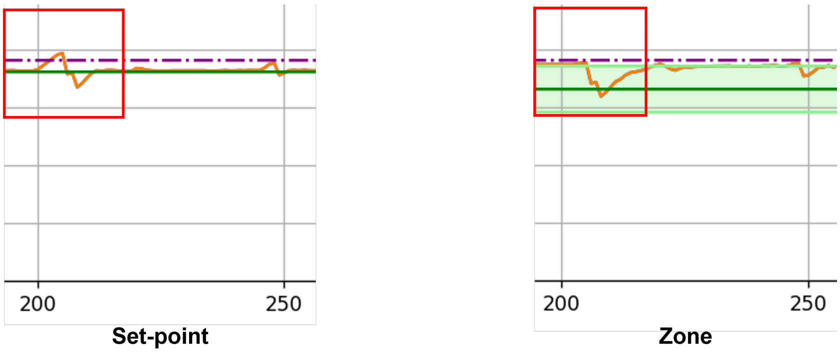

Figure 4.8: Set-Point MPC vs Zone MPC

4.5. Online Model Estimation - Evaluation

In this section, the effect of integrating online model learning into the Zone MPC controller is evaluated. The aim is to assess whether the RLS filter can learn the water balance model coefficients online, eliminating the need for historical data and prior knowledge. Historical data was previously also used to model ET_c, the evapotranspiration of a specific crop over a growing season. In this section, we will also evaluate the effect of using ET_o instead of ET_c as an input to the RLS filter. As introduced in Equation 3.12, this means the filter must track the combined coefficient $\bar{c_2}$, which absorbs the seasonal variation in K_c. Whether the RLS can do this effectively is evaluated in subsection 4.5.2

The controller in this section uses the same tuning of the Zone MPC as in the previous simulations (see Table 4.1). The Recursive Least Squares filter is tuned through trial and error (see Table 4.3). All model coefficients are initialised to 0.5, with their signs determined by physical reasoning. For example, rainfall and irrigation have a negative sign, since increased irrigation reduces the RZD. Each coefficient is also given an upper and lower bound to prevent unstable behaviour if early measurements cause the sign of a coefficient to flip.

Table 4.3: RLS Simulation Parameters

Parameters	Values
Forgetting factor: λ	0.99
Initial Covariance: $P(0)$	$0.1I_{4\times4}$
Initial Parameter Vector: $\theta(0)$	[0.5, 0.5, -0.5, -0.5]
Upper bound on θ:	[2, 2, -0.2, -0.2]
Lower bound on θ:	[0.2, 0.2, -2, -2]

4.5.1. Online Learning using ET_c

This subsection evaluates the case in which the RLS filter has access to ET_c as input. This serves as an intermediate step: the online learning mechanism is tested while the seasonal variation in crop water demand is still provided, isolating the effect of learning the remaining model coefficients online.

A simulation was run for the 2015–2016 growing season, assuming that RZD measurements are available every day (i.e., no sparsity), and assuming knowledge of ET_c. The results of this RLS adaptation are compared with those of the Zone MPC simulation in subsection 4.4.4, as both use the same cost function and planting date. Figure 4.9 shows the two simulations side by side. The RZD in both follows a similar overall trajectory, although some noticeable differences can be observed. During the first 50 days of the growing season, substantial precipitation occurs, leading both controllers to decide not to irrigate and resulting in a similar RZD trajectory.

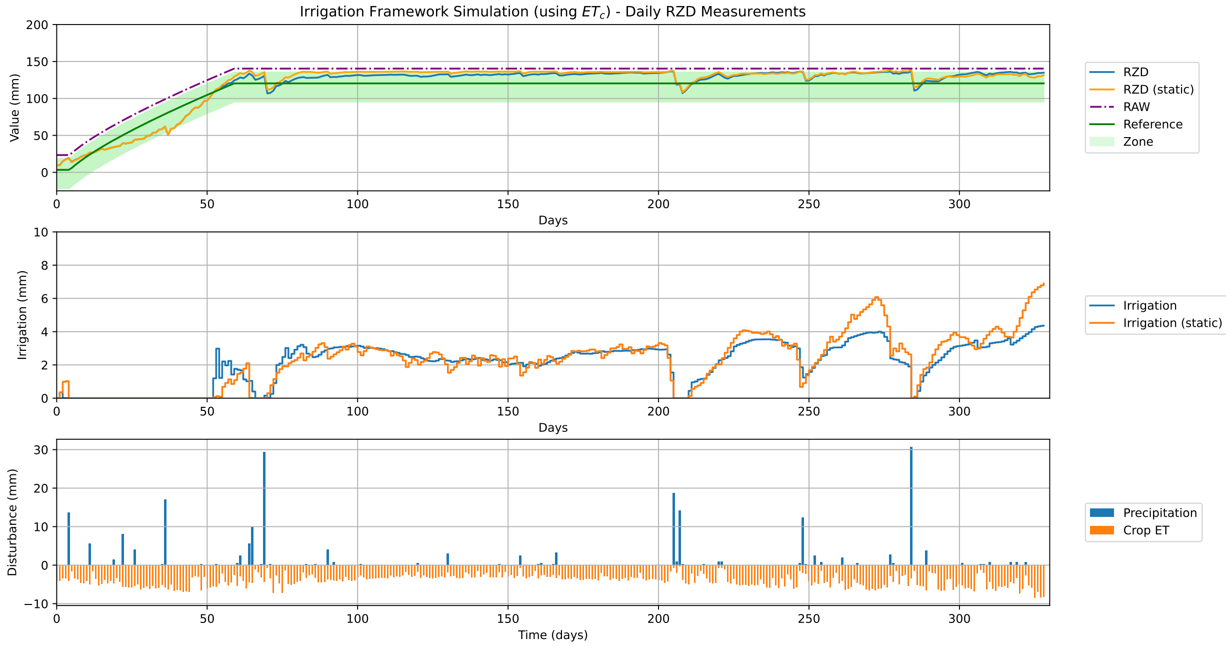

Figure 4.9: Zone MPC simulation (2015-2016), with online learning (using $\mathbf{ET_c}$) vs Static Model

From day 50 to 200, it can be seen that the Zone MPC + RLS does not stay as close as possible to the upper edge of the zone, which is unexpected given the formulation of the cost function. In contrast, the RZD of the Zone MPC without RLS moves closer to the upper bound of the zone. This behaviour is also visible in the irrigation plot, where the irrigation signal fluctuates much more, as it is operating closer to the edge. The irrigation profile of the RLS-based controller is considerably smoother because it operates less at the zone boundary. Even though one strategy shows more peaks and drops in irrigation while the other follows a smoother trajectory, the total irrigation usage over this period should be roughly the same for both controllers.

From day 200 onward, the irrigation applied by the Zone MPC is substantially higher during certain periods (e.g., days 260-280). This observation suggests that the Zone MPC + RLS has a better understanding of the underlying system dynamics in this phase. Another indication of this is that, toward the end of the season, the RLS-based controller is better able to stay near the upper edge of the zone, as expected given the controller's cost function. This behaviour is due to the RLS filter capturing changes in the crop's response to irrigation and adapting accordingly. This adaptation is illustrated in Figure 4.10, which shows the evolution of the model coefficients identified by the RLS over the growing season. From day 160 until the end of the season, c_4 (corresponding to irrigation) decreases from -0.8 to around -1.3, whereas the coefficient identified for the Zone MPC remains at -0.82.

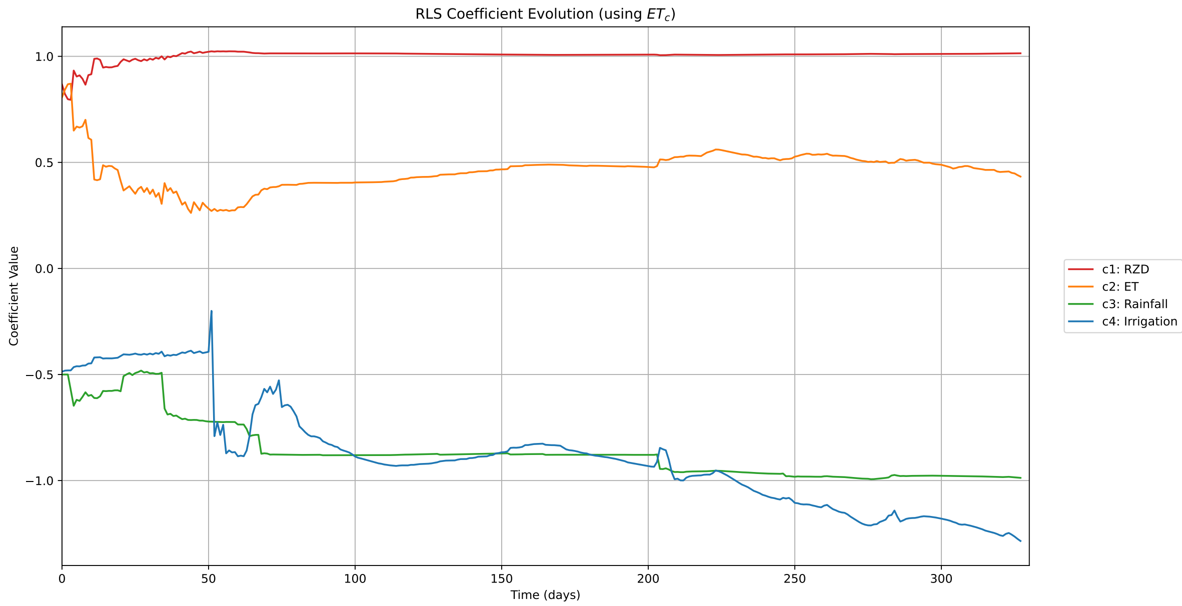

Figure 4.10: Coefficient Evolution of Zone MPC + RLS using $\mathbf{ET_c}$ (2015-2016)

If we examine the evolution of the coefficients further, we see that during the first 50 days, c_4 slowly drifts towards -0.4, this is the opposite of what one would expect. This effect happens because there is a lot of rainfall early in the season, meaning the controller does not need to irrigate to maintain the desired soil moisture. Without irrigation events, the RLS filter cannot accurately assess the effect of irrigation on the RZD, leading to fluctuations in c_4. Although this coefficient does not accurately reflect the actual effect of irrigation on the crop, it does not pose a practical issue, since once irrigation starts, the controller can quickly compensate for any resulting error. This behaviour is visible around day 52, the RLS-based controller resumes irrigation. Initially, it applies slightly too much irrigation due to a poor estimate of the irrigation effect, but the RLS adapts quickly to correct this error. The soil dynamics are quite forgiving in this respect, as small missteps by the controller can be corrected over the following days.

The coefficient c_1 quickly converges close to 1, as expected, since it represents the contribution of the soil state from the previous day. The filter isolates this effect quite clearly. The trajectory of c_2 also behaves as expected. In this simulation, the RLS filter receives ET_c as an input, so we expect c_2 to remain relatively constant. This is because the seasonal variation in the effect of evapotranspiration on the crop is already captured by the crop coefficient K_c (see Equation 2.6).

The performance metrics (see Table 4.4) show that the Zone MPC + RLS achieves the same fresh yield, while w_p improves slightly. This increase is a positive result, indicating that the crop experienced less water stress. The smaller amount of irrigation applied by the RLS-based controller during the second half of the season, while maintaining the same yield, results in a substantially higher $IWUE$ than the Zone MPC. With WUE also being slightly higher, the performance metrics indicate that, for this season, the Zone MPC + RLS using ET_c performs better than the Zone MPC with a model based on historical data.

4.5.2. Online Learning using ET_o

As in the previous section, the 2015–2016 season is simulated again. However, in this case, the controller is not given access to ET_c; instead ET_o is used as an input to the RLS filter. Unlike ET_c, ET_o only represents the atmospheric demand for water and does not account for the crop-specific response. As before, the results are compared with the (static) Zone MPC controller from section 4.4 to highlight the differences.

Figure 4.11 shows the two controllers side by side. The behaviour is in many ways similar to the previous simulation, where the RLS filter used ET_c. At the start and during the middle of the season, the irrigation patterns are quite similar, and the Zone MPC behaves as expected. For example, the controller still reduces irrigation when a precipitation event is forecast. The main difference from the previous case is that the RLS-based controller no longer clearly outperforms the static controller, as it no longer irrigates less. The total irrigation applied is now at a similar level, suggesting that the controller no longer has a better estimate of the crop's water demand. This becomes apparent toward the end of the season. In this phase, the controller behaves similarly to the static controller, assuming that more irrigation is required than is actually necessary, which leads to over-irrigation.

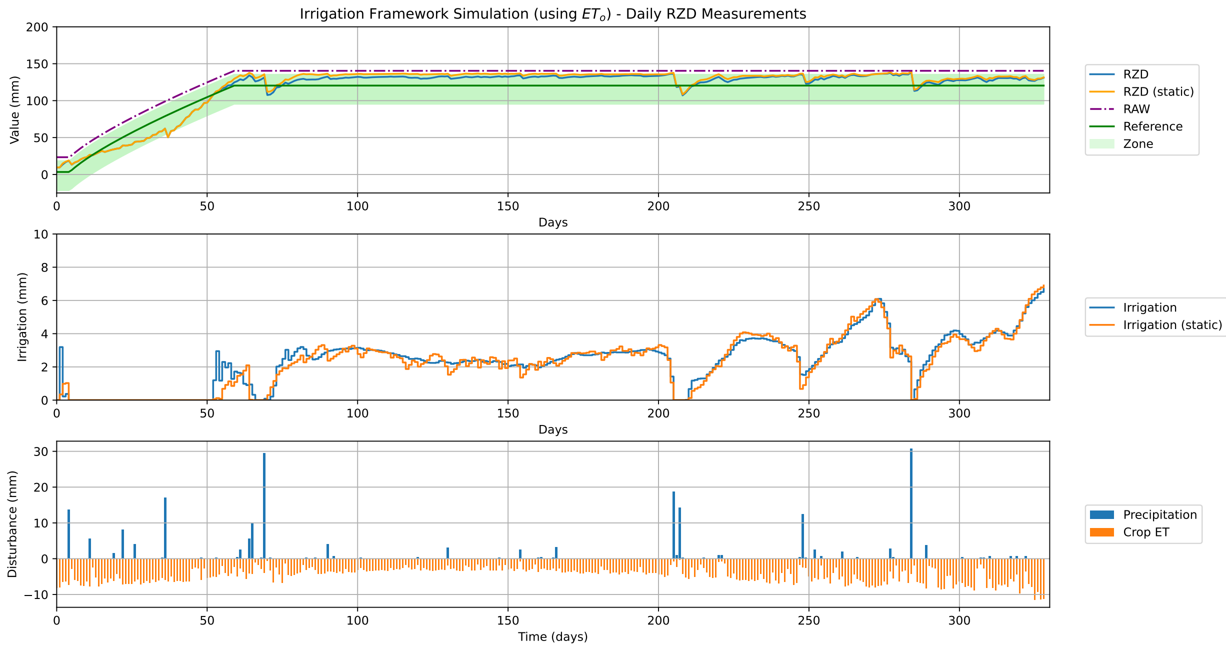

Figure 4.11: Zone MPC simulation (2015-2016), with online learning (using $\mathbf{ET_o}$) vs Static Model

If we consider the evolution of the coefficients in Figure 4.12, the picture is notably different from the ET_c case. The coefficients c_1, c_3 and the early behaviour of c_4 are broadly similar. The differences lie in $\bar{c}_2$ and the late-season behaviour of c_4. Throughout the season, $\bar{c}_2$ increases slowly from approximately 0.35 to 0.6, the RLS consistently underestimates the effect of evapotranspiration on the RZD. The slow upward drift does not reflect successful tracking of K_c, as it never approaches the true value.

It is worth noting that in the ET_c case, c_2 also sits well below the identified value of 0.95, remaining around 0.5 for most of the season. However, performance there is good. This shows that the level of underestimation alone does not determine performance. The difference is whether the underestimation is constant or time-varying. In the ET_c case, because K_c is already embedded in the ET_c signal, the underestimation of c_2 is roughly constant throughout the season. The other coefficients, particularly c_4, can stably compensate for this, and the system reaches a consistent equilibrium. In the ET_o case, however, the true value of $\bar{c}_2 = c_2 \cdot K_c$ itself varies over the season as K_c evolves. The degree of underestimation, therefore, changes over time, meaning the compensation required from the other coefficients also needs to shift throughout the season.

This is reflected in the behaviour of c_4. In the ET_c case, c_4 increases in magnitude from -0.8 to approximately -1.3 from day 200 onward, reflecting a learned improvement in understanding the irrigation dynamics in the late season. In the ET_o case, c_4 remains between -0.8 and -1.0 and never reaches this magnitude. The compensation it provides is partly because of the need to offset the time-varying ET underestimation, which prevents it from capturing the same late-season improvement. The result is that total irrigation ends up similar to the static model, but the late-season gain visible in the ET_c case does not appear.

The cause of the RLS failing to track $\bar{c}_2$ reliably has to do with the nature of the ET_o input signal. Irrigation and rainfall have attributable effects on RZD, providing the RLS with an unambiguous update signal. ET_o, by contrast, is present every single day as a smoothly varying non-zero value. It never switches off or produces a sharp effect, making it much harder for the RLS to isolate its specific contribution to the prediction error. Combined with the fact that corrections are distributed across all coefficients simultaneously rather than being cleanly attributed to $\bar{c}_2$, the RLS cannot reliably detect and correct the growing mismatch as K_c evolves over the season.

A dedicated online K_c estimator driven by canopy cover measurements, discussed in section 6.1, could address this limitation while preserving the design philosophy of the framework.

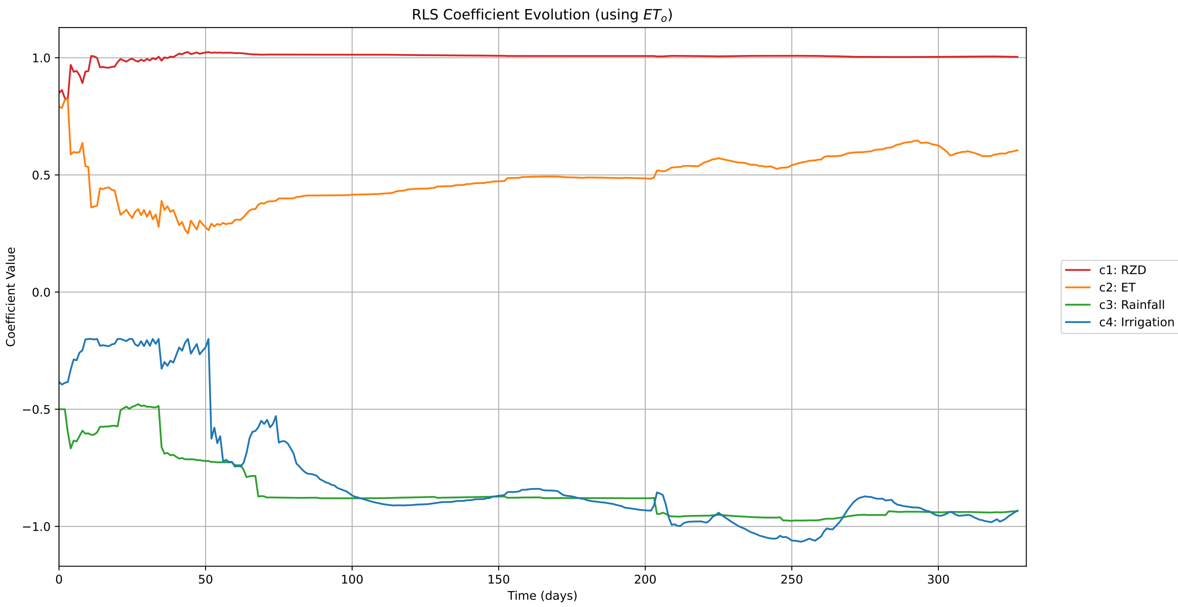

Figure 4.12: Coefficient Evolution of Zone MPC + RLS (using $\mathbf{ET_o}$ (2015-2016)

Having said all that, if we consider the performance metrics in Table 4.4, it can be seen that the simulated controller in this section still has the same w_p, indicating that the crop does not undergo extra water stress. However it performed more unnecessary irrigation than the RLS based controller utilising ET_c. This is reflected in the lower $IWUE$ and WUE. Compared with the Zone MPC with a static model, performance is roughly equal across all metrics, with a marginally better $IWUE$. Although it is less accurate than the version using ET_c, this controller can learn the model online with no prior knowledge and achieve performance comparable to a static MPC that relies on historical data and ET_c.

Irrigation Strategy	$w_p(kg/m^3)$	$IWUE(kg/m^3)$	WUE $(0-1)$	Fresh Yield (kg/m^3)
1) Zone MPC (Static Model)	8.492	10.613	0.943	8,063
2) Zone MPC + RLS with ET_c	8.826	11.425	0.959	8,058
3) Zone MPC + RLS with ET_o	8.540	10.733	0.955	8.418

Table 4.4: Performance Metrics - Zone MPC + RLS Simulation (2015-2016)

4.6. RAW Estimator - Evaluation

Before combining all components, the RAW estimator from section 3.5 is evaluated in isolation. The goal is to determine how accurately the piecewise linear rooting-depth approximation tracks the true RAW trajectory and how many Z_r measurements are needed to achieve sufficient accuracy. One important aspect to consider is the interval between successive Z_r measurements $k_{meas} - k_{prev}$, since shorter intervals naturally increase the total number of measurements in a growing season.

The RAW estimator was configured with a stagnation threshold of 0.005, meaning that when G_r falls below this value, the crop is assumed to have reached its maximum rooting depth. The chosen measurement strategy measures Z_r every 3 days during the first 15 days, after which measurements are taken every 7 days until growth stagnation is detected. This approach is intended to capture the onset of root growth as early as possible. The RAW estimator was applied to the same 2015–2016 growing season, and the evolution of the growth ratio is shown in Figure 4.13. The red dots indicate measurement days; on those days, the slope of the green line represents the estimated growth rate. Each line spans 10 days, corresponding to the prediction horizon. It is important to note that on days between measurements, the estimator continues to extrapolate the most recently estimated growth ratio, which can lead to increasing deviation from the true RAW. The most significant deviations occur at the beginning and the end of the growth curve. Early in the season, these deviations result in an underestimation of rooting depth, which may lead to over-irrigation because the soil has a higher water-holding capacity than expected. Toward the end of root development, the opposite occurs: Z_r is temporarily overestimated, which could lead to short periods of water stress. The figure only shows the first 80 days of the season because, for sugarcane, root growth stops after this period.

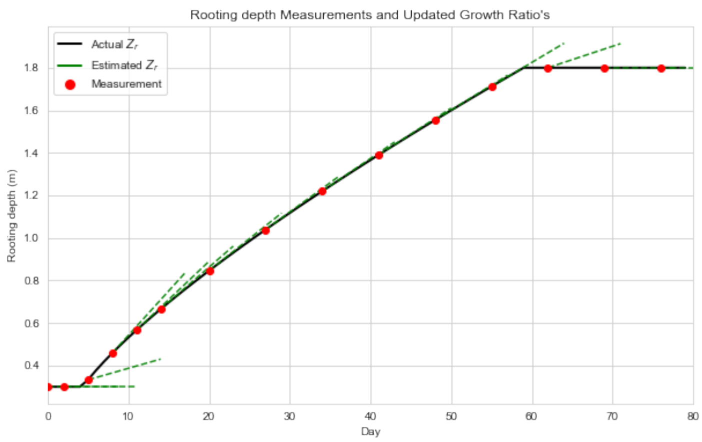

Figure 4.13: Estimated Z_r trajectory for Sugarcane (2015-2016)

The RAW is computed from the estimated Z_r. Figure 4.14 compares the predicted RAW trajectory over the growing season with the true RAW. Similar to the Z_r trajectory, a slight deviation is visible at the start of the season, along with an overshoot around day 60. If the maximum rooting depth Z_{max} is assumed to be known, this overshoot disappears, as shown in Figure 4.15.

Overall, this simple piecewise linear method for RAW estimation appears sufficiently accurate to serve as a reference for the irrigation controller. While there are two periods during the season where the RAW estimate deviates from the actual value, the fact that only 15 Z_r measurements are required here makes this a reasonable trade-off. The next step is to examine how these estimation errors propagate when the RAW estimator is fully integrated with the RLS filter and the Zone MPC.

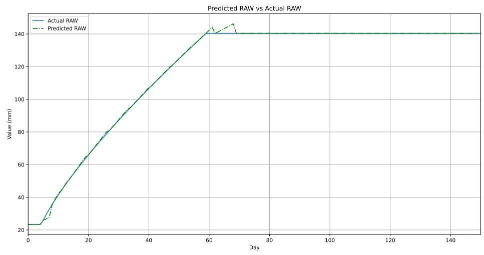

Figure 4.14: Estimated RAW trajectory (2015-2016)

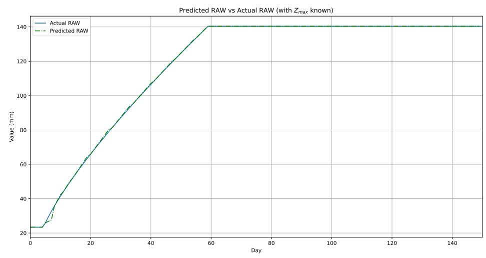

Figure 4.15: Estimated RAW trajectory, with $Z_{max} known$ (2015-2016)

4.7. Integrated Framework - Evaluation

In the previous sections, we evaluated the individual components separately to assess their behaviour and impact. In this section, all components are combined into a single integrated framework, following the architecture described in section 3.2. The aim is to assess the framework's overall strengths and weaknesses and determine whether it can achieve its intended goals within the given constraints. The predictive framework is designed to learn its model parameters in real time, without relying on prior data or crop-specific information. It should be able to adapt to changes in ET's effect on the crop, since no modelled ET_c is available beforehand. The Zone MPC is expected to keep the RZD within the desired bounds while anticipating future water demand, using an estimated RAW trajectory over the prediction horizon as a reference. This RAW trajectory is provided by the RAW estimator, which relies on rooting depth measurements.

The framework is evaluated in two different configurations. First, a scenario with daily RZD measurements is implemented, in which the Zone MPC receives measurements each day and the RLS performs daily updates. Second, the sparse RZD measurement strategy is implemented to examine how the framework performs with fewer data points and less frequent updates. In both configurations, the integrated framework is evaluated using ET_o, since the goal of this thesis is to assess its capabilities without relying on prior information. In earlier sections, where the individual components of the framework were evaluated separately, the experiments used 2015–2016 growing-season data. In this section, the performance metrics of the proposed framework are calculated across all available seasons. In addition, two other growing seasons are shown to illustrate how weather conditions can vary from year to year.

4.7.1. Daily RZD measurements

In this section, the performance of the irrigation framework is evaluated under the assumption of daily RZD measurements. First, the controller's behaviour is analysed for a single growing season by

Table 4.5: Simulation Parameters - Predictive Online Irrigation Framework

Parameters	**Values**
Prediction Horizon (N)	10
Input Cost (R)	100
State cost (Q_{upper})	100
State cost (Q_{lower})	1
Zone size (l_{upper})	15
Zone size (l_{lower})	25
Reference offset	20
RLS: Forgetting factor: λ	0.99
RLS: Initial Covariance: $P(0)$	$0.1 I_{4\times4}$
RLS: Initial Parameter Vector: $\theta(0)$	[0.5, 0.5, -0.5, -0.5]
RLS: Upper bound on θ:	[2, 2, -0.2, -0.2]
RLS: Lower bound on θ:	[0.2, 0.2, -2, -2]

examining the decisions it makes. Then, the performance metrics are evaluated across all growing seasons from 2013 to 2018.

In Figure 4.16, the simulation results of 2016-2017 are shown. The blue lines in the top and middle figures represent the RZD and the irrigation applied in the proposed framework. The orange lines correspond to a Zone MPC controller with a static model, as described in subsection 4.4.1, that uses ET_c instead of ET_o and a reference based on historical data. Looking at the overall season, the Zone MPC component in both controllers behaves as expected. For example, both anticipate upcoming rainfall and reduce irrigation in advance, as shown around days 230 and 275. In general, the RZD and irrigation trajectories of both strategies are similar, although there are some notable differences.

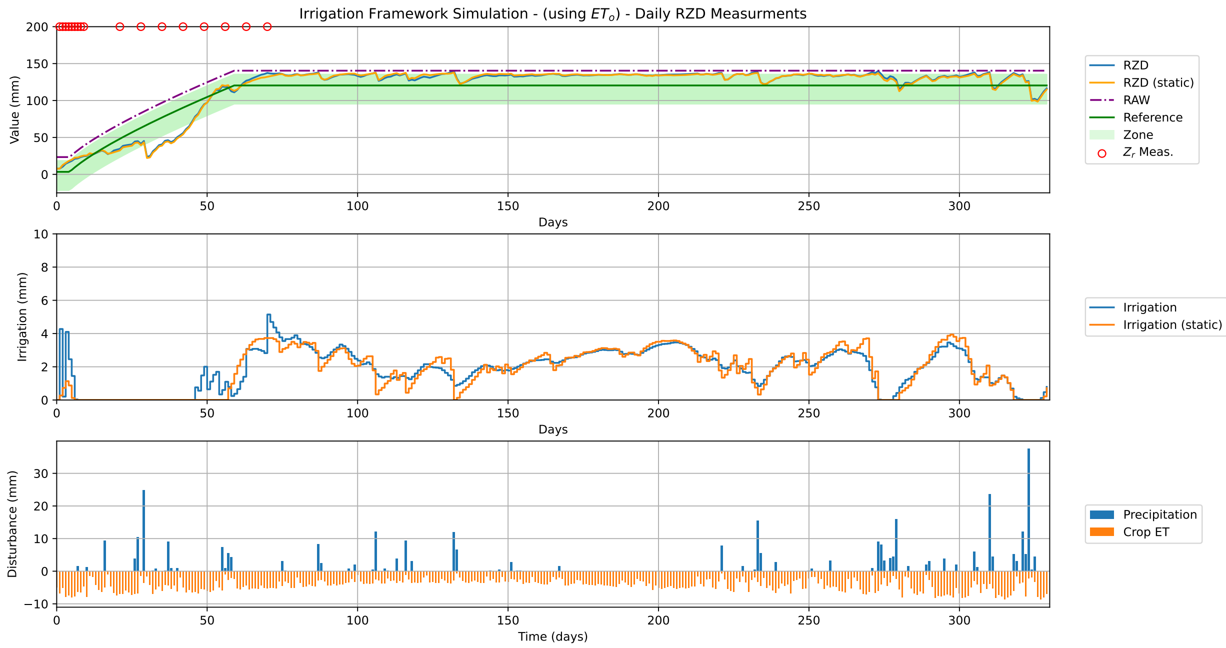

Figure 4.16: Irrigation Framework Simulation (2016-2017) - Daily RZD Measurements

One such difference appears at the beginning of the season, where the proposed framework applies more irrigation on two days compared to the static-model controller. This behaviour occurs because the RLS component still needs to refine its estimate of the effect of irrigation on soil moisture. Since the framework starts with an initial guess of $c_4 = 0.5$, it initially over-irrigates. However, since the initial covariance $P(0)$ of the RLS is still relatively high, the framework can update c_4 to a value that better reflects the soil–crop dynamics at that time. Essentially, there is a trade-off at the start of the season: a small amount of over-irrigation is accepted in order to obtain a more accurate estimate of the model coefficients.

The framework was also tested with a higher initial covariance matrix of $10I_{4\times4}$. In that case, the over-irrigation at the start of the season is much smaller, since the filter performs larger parameter updates. However, this comes with the drawback that $P(k)$ decays rapidly. Within the first few days, it becomes very small, and after these initial large updates, it takes a long time for it to recover to a meaningful magnitude. As a result, the RLS filter effectively stops learning after the first couple of weeks, which is undesirable. There is a trade-off between early speed and long-term adaptability.

The red circles at the top of Figure 4.16 indicate the days on which rooting depth measurements are taken. The selected measurement strategy for the framework is to measure Z_r daily for the first 10 days, then every 7 days thereafter. This differs slightly from the strategy evaluated for the RAW estimator in section 4.6. The adjustment was made because it is crucial to detect the onset of root growth and its initial growth rate as early as possible. If the start of growth is detected even two days too late, this can significantly affect the predicted RAW over the MPC prediction horizon during those days. As mentioned earlier, crop rooting depth generally follows a sigmoid logistic curve over time [20]. Around the inflexion point, the growth is approximately linear. This means that after the initial growth phase, measuring once every 7 days is sufficient. Measurements continue until growth stagnation is detected, resulting in a total of 17 Z_r measurements. As shown in section 4.6, the most significant deviations in RAW prediction occur at the beginning and end of the growth curve. In this simulation, however, these deviations are not clearly visible because rainfall prevents the RZD from approaching the edge of the target zone closely enough to trigger different irrigation decisions. The reference shown in the figure corresponds to the predetermined reference used by the static Zone MPC controller.

From day 70 to day 200, the RZD of the proposed framework does not stay as close to the upper edge of the zone as the static Zone MPC controller does. The latter behaviour is consistent with the formulation of the Zone MPC cost function. The proposed framework also struggles with adapting as quickly to upcoming precipitation events, in the mid-season. From day 200 onward the proposed irrigation framework is better able to time its irrigation events and operate closer to the upper zone boundary. These observations align with the findings in section 4.5. The static Zone MPC controller appears to capture the system dynamics more accurately during the middle of the season. The proposed framework improves it's tracking capabilities towards the end of the season.

In section 4.5, it was shown that when the RLS filter uses ET_c as an input, the performance improves significantly. A similar effect would occur here. In that case, the proposed framework would outperform the static model toward the end of the growing season and would also achieve better accuracy during the mid-season. The RLS component would then be able to adapt to changing dynamics in the final part of the season, leading to more efficient water use. In contrast, the static model relies on fixed coefficients that are generally suitable for most of the season but cannot adapt to changing conditions. In the current setup, the framework instead uses ET_o, which introduces a time-varying error. The differences between using ET_c and ET_o are discussed in more detail in section 4.5.

After evaluating the framework for a single growing season, the next step is to assess the performance metrics (see section 4.3) across all available weather data from 2013 to 2018. This is important because results from a single season may give a biased impression of performance. Weather conditions vary significantly between seasons, which directly affects how the controller performs. Table 4.7 presents the performance metrics of the proposed irrigation framework, while Table 4.8 shows those of the static Zone MPC introduced earlier in this section. The static controller serves as a useful benchmark, since metrics such as WUE can vary substantially between seasons due to differences in precipitation, for example, one season may receive twice as much rainfall as another. We also provide the performance metrics of the integrated framework, had it used ET_c, in Table 4.6

Table 4.6: Performance Metrics - Irrigation Framework with daily RZD measurements using ET_c

Start date	w_p (kg/m^3)	$IWUE$ (kg/m^3)	WUE	Irr (mm)	Fresh Yield (kg/m^3)
10-02-2013	8.894	20.228	0.745	426	8.614
10-02-2014	8.817	13.144	0.578	630	8.283
10-02-2015	8.701	11.065	0.940	700	7.744
10-02-2016	8.941	13.916	0.849	595	8.278
10-02-2017	8.908	18.508	0.845	464	8.588

Table 4.7: Performance Metrics - Irrigation Framework with daily RZD measurements using ET_0

Start date	w_p (kg/m^3)	$IWUE$ (kg/m^3)	WUE	Irr (mm)	Fresh Yield (kg/m^3)
10-02-2013	8.883	19.610	0.743	444	8.709
10-02-2014	8.784	12.898	0.581	651	8.397
10-02-2015	8.532	10.796	0.948	738	7.963
10-02-2016	8.910	13.535	0.843	618	8.371
10-02-2017	8.884	18.028	0.842	480	8.647

Table 4.8: Performance Metrics - Zone MPC with Static Model

Start date	w_p (kg/m^3)	$IWUE$ (kg/m^3)	WUE	Irr (mm)	Fresh Yield (kg/m^3)
10-02-2013	8.887	20.089	0.740	426	8.556
10-02-2014	8.755	12.886	0.580	647	8.342
10-02-2015	8.492	10.613	0.943	760	8.063
10-02-2016	8.849	13.720	0.850	602	8.259
10-02-2017	8.869	18.006	0.843	480	8.644

Overall, the results of the irrigation framework using ET_0 and the controller with a static model are relatively similar across all seasons. The average $IWUE$ of the framework across the five seasons is lower by about 0.09, a small difference. Most of this difference comes from the 2013 season, where the static controller outperforms the framework by 0.48. Coincidentally, the training data used to build the static model also comes from this same growing season, which likely makes the static model relatively more accurate in that case. In contrast, the framework achieves a higher $IWUE$ than the static controller in 2014 and 2015. These differences in $IWUE$ can be explained by the behaviour observed earlier. The proposed framework irrigates somewhat less effectively during the mid-season but performs better toward the end of the season compared to the static Zone MPC. For example, in 2013, a large amount of rainfall occurred near the end of the growing season, to the point that no irrigation was needed during that period. As a result, the framework could not benefit from its improved late-season performance and was unable to compensate for its relatively weaker mid-season decisions. In contrast, the conditions in 2014 and 2015 allowed the framework to leverage its better late-season adaptation, resulting in a higher $IWUE$ than the static Zone MPC. To put differences in $IWUE$ into more practical terms, consider a controller with an $IWUE$ that is $0.1kg/m^3$ higher with a seasonal irrigation of 600 mm. On one hectare of land this difference corresponds to approximately 60,000 litres of water saved compared to its counterpart.

When looking at the framework using ET_c, the $IWUE$ is, as expected, higher in every season compared to the version using ET_o, but also consequently outperforms the static controller. In this case, the controller does not need to fully learn the effect of evapotranspiration, resulting in better overall performance compared to the static controller.

It is also worth noting that w_p remains relatively constant across all seasons in the proposed framework, both with ET_c and ET_o. Large deviations in this metric indicate that the crops experience significant periods of water stress. Keeping this value relatively constant is important, since allowing more water stress would be an easy way to increase irrigation efficiency, but would come at the expense of crop health and yield.

4.7.2. Sparse Measurement Strategy

In this section, the integrated framework is evaluated using the sparse measurement strategy introduced in subsection 3.6.2. In short, this strategy combines interval-based measurements, where the interval increases as the season progresses, with event-triggered measurements for precipitation and irrigation, which are activated if no measurement has been taken for τ consecutive days after such an event. For this test setup, the parameters were set to $f_{start} = \frac{1}{2}$ and $f_{end} = \frac{1}{10}$ for the "heartbeat" measurements, and $\tau = 10$ for the event-triggered measurements.

With fewer RZD measurements, the framework has access to fewer updates and less information. This section aims to assess whether the proposed framework can still perform adequately under this additional limitation. As in the previous section, one growing season is first analysed in detail, followed by an evaluation of the performance metrics across all available seasons.

Figure 4.17 shows a simulation using this measurement strategy, presented in the same format as in the previous section. The blue 'x' symbols at the bottom of the top figure indicate the days on which RZD measurements are taken. As expected, measurements are taken more frequently at the beginning of the season, with the interval between measurements gradually increasing over time. In some cases, measurements are taken on consecutive days due to event-triggered updates. Overall, the controller's behaviour under this measurement strategy is similar to what was observed with daily RZD measurements. Heavy rainfall early in the season causes the RZD to exceed its lower bound, and an extreme precipitation event around day 80 leads to a similar effect. These violations are acceptable, as they concern the lower boundary and are caused by rainfall rather than inadequate irrigation. At the end of the growing season, we can see that the framework using the sparse measurement strategy performs similarly to the one with daily measurements. The slight model mismatch observed in the previous section with daily measurements persists. However, it does not appear to worsen significantly, even as the frequency of RZD measurements decreases toward the end of the season. The event-driven measurement component of the strategy focuses on capturing irrigation and rainfall events, helping the framework maintain a reasonable estimate of the corresponding model coefficients.

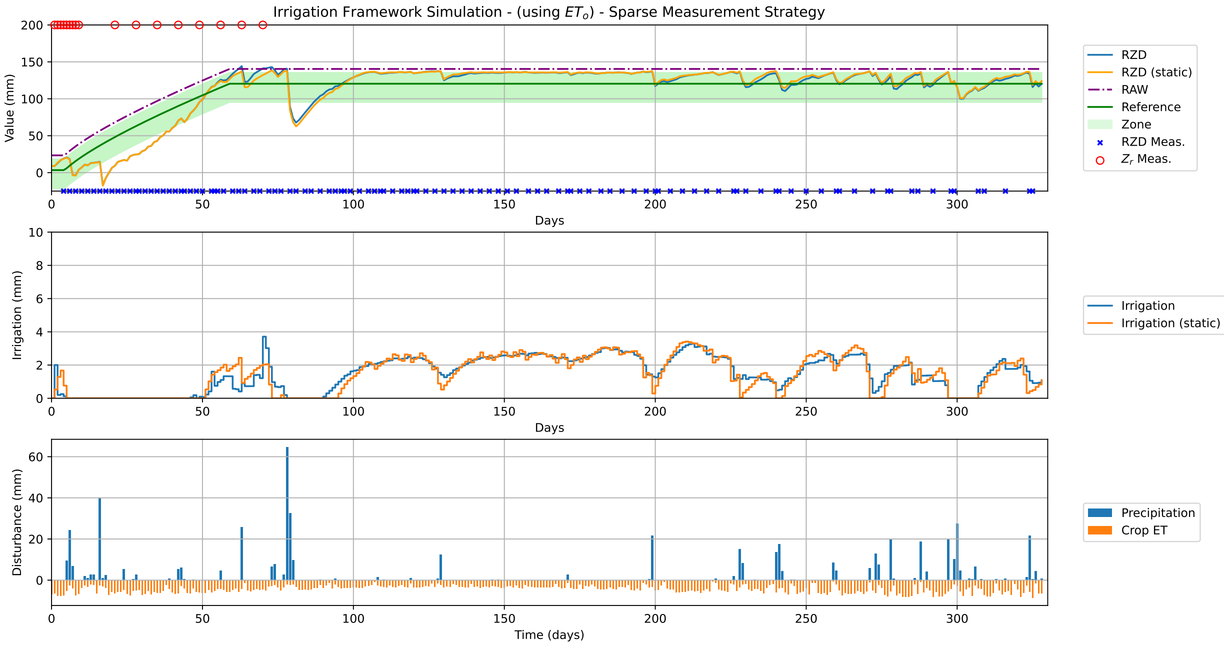

Figure 4.17: Irrigation Framework Simulation (2017-2018) - Sparse Measurement Strategy

Table 4.9 shows the performance metrics across the growing seasons from 2013-2018 for the framework using the sparse measurement strategy. In general, every metric, across the board, is similar if we compare it to Table 4.7. The $IWUE$ compared to the proposed framework using ET_o with daily measurements is, on average across the five seasons, only 0.05 lower. At the same time, the amount of RZD measurements required has been reduced by about 65%. The exact number of measurements differs from season to season. This concerns event-triggered measurements; the number of 'heartbeat' measurements is the same season to season. Overall, the proposed irrigation framework with the sparse measurement strategy based on ET_o performs almost as well as the version using daily measurements, indicating that the sparse strategy is effective.

Table 4.9: Performance Metrics - Full Framework with Sparse Measurement Strategy - Using ET_o

Start date	w_p (kg/m^3)	$IWUE$ (kg/m^3)	WUE	Irr (mm)	Fresh Yield (kg/m^3)	RZD Meas. (#)
10-02-2013	8.880	19.651	0.742	441	86.698	108
10-02-2014	8.753	12.697	0.581	665	84.402	107
10-02-2015	8.688	11.002	0.940	713	78.427	108
10-02-2016	8.896	13.442	0.845	630	84.639	112
10-02-2017	8.870	17.792	0.841	488	86.849	113

5
Discussion

In this chapter we reflect on the performance of the proposed control framework by discussing the results, the assumptions that have been made, and some of the practical implications.

5.1. Performance of the Proposed Framework

The evaluation in Chapter 4 was carried out in stages, first assessing each component separately before combining them into the full framework. This approach makes it easier to see where performance gains or losses occur, and to understand the behaviour of the integrated system. The discussion below follows the same structure.

- Starting with **the Zone MPC using a static model**, the controller performs well in this configuration. It can keep the RZD within the target zone and make sensible use of weather forecasts, reducing irrigation before rainfall events. The comparison with the FCSI controller showed that the Zone MPC does not expose the crop to more water stress. The comparison between Zone MPC and a traditional set-point MPC is also informative. The set-point formulation penalises deviations on both sides of the reference equally, so the controller actively irrigates to move the RZD back toward the reference when it drops below it. In an irrigation context, a lower RZD means the soil holds more water, which does not immediately harm the crop. **The Zone MPC** only penalises deviations that exceed the zone boundaries, which means it **does not waste water trying to maintain an unnecessarily precise soil moisture level**. The set-point control strategy can lead to unnecessary water stress on the crop, whereas Zone MPC avoids it.
- When considering **online model estimation using $\mathbf{ET_c}$**, the RLS filter learns reasonable coefficient values within a few weeks and adapts meaningfully as the season progresses. The most consequential adaptation occurs from approximately day 160 onwards, when the irrigation coefficient c_4 shifts significantly, reflecting a change in the crop's actual response to irrigation that the static model cannot capture because it holds all coefficients fixed throughout the season. **This late-season adaptation leads to measurably better performance across the water use metrics compared to the static controller, while fresh yield remains the same.** The performance is not better across the entire season. In the mid-season, the RLS-based controller operates slightly further from the zone boundary than the static controller, but it compensates by using water more

efficiently toward the end. One limitation that arises early in the season is that during periods of heavy rainfall, when irrigation is rarely applied, the filter lacks sufficient input to accurately estimate the irrigation coefficient c_4, leading it to temporarily drift in the wrong direction. However, once irrigation continues, the filter quickly corrects itself. The forgiving dynamics of the soil system mean that early estimation errors do not accumulate into lasting performance degradation, and the controller recovers within a few days.

- When considering **Online model estimation using** ET_0, the filter struggles to track the seasonal evolution of crop water demand. In this configuration, the filter is implicitly expected to absorb the seasonal variation in K_c into the combined coefficient $\bar{c}_2$. In practice, this proves unreliable, likely because ET_0 is present every single day as a smoothly varying non-zero value. Unlike irrigation or rainfall, it never switches off or produces a sharp, attributable effect on the RZD. This makes it much harder for the RLS to isolate its specific contribution to the prediction error. Combined with the fact that the corrections are distributed across all coefficients simultaneously rather than only being attributed to $\bar{c}_2$, the filter cannot reliably detect and correct the growing mismatch as K_c evolves over the season. The result is that the ET coefficient does not clearly follow the expected seasonal trajectory. **Despite this limitation, the ET_0-based controller still performs comparably to the static model.** The static model requires a historical dataset and a pre-computed crop coefficient curve to achieve similar performance, whereas this configuration requires neither.
- **The RAW estimator**, based on a piecewise linear approximation of rooting depth, provides a sufficiently accurate reference signal for the Zone MPC while requiring only about 15 rooting depth measurements over the entire growing season. The estimator shows two periods of deviation: an underestimation early in the season when root growth is first detected, and a brief overestimation near the end of the growth phase. The first can lead to slight over-irrigation, since the soil's actual water-holding capacity is larger than estimated, while the second introduces a small risk of short-term water stress. **Given the low number of required measurements, this is a reasonable trade-off** for a framework that does not rely on crop-specific knowledge.
- In the **integrated framework with daily measurements**, when all three components are combined and evaluated across five growing seasons from 2013 to 2018, the results are generally consistent. The w_p metric remains stable across all seasons for the proposed framework, confirming that the controller does not sacrifice crop health for water savings. The $IWUE$ varies between seasons, as expected, since rainfall patterns directly affect the achievable irrigation efficiency regardless of the control strategy. **Comparing the integrated framework (using ET_0) to the static Zone MPC, the performance gap is small in most seasons.** The framework achieves a higher IWUE in seasons where late-season conditions allow it to leverage its adaptive learning. It falls slightly behind in seasons where the static model's mid-season accuracy dominates. Across the full five-season evaluation, no strategy consistently dominates, and the performance difference is small enough that the framework's ability to operate without prior data represents a practical advantage. **The version of the framework using ET_c, outperforms the static controller in every season,** confirming that the RLS adaptation mechanism functions correctly when given the right input signal.
- The **sparse measurement strategy** replaces daily RZD measurements with a combined heartbeat-and-event-triggered approach, reducing the total number of soil moisture measurements by approximately 65%, from roughly 330 daily observations to 108-113 per season. Despite this substantial reduction in measurements, the performance metrics across all seasons remain almost unchanged. The average $IWUE$ difference compared to the daily-measurement version

is negligible, and the w_p values remain stable, indicating no increase in crop water stress. Two properties of the framework explain this robustness. First, the receding-horizon structure of the Zone MPC provides resilience to occasional state estimation errors. Because the optimisation is re-solved each day using the best available information, the controller can recover from periods without measurements. Second, the event-triggered component of the measurement strategy ensures that the most informative observations, those immediately following irrigation or rainfall events, are prioritised. These are precisely the moments when the RLS filter needs new data to maintain accurate coefficient estimates for irrigation and precipitation, and the measurement strategy is designed to capture them.

- An **overall assessment** of the results shows that the proposed framework achieves its main objective: effective predictive irrigation control without relying on historical data or crop-specific models. The performance gap compared to a configured static controller is small, and in some cases, the framework even outperforms it. The main limitation is the RLS filter's inability to track the seasonal evolution of K_c when only ET_o is available, which limits how effectively the framework can leverage its learning capability. Addressing this, for example, by introducing a dedicated online K_c estimator, could further improve performance. **The results with the sparse measurement strategy further suggest that the framework is practically deployable, as the required measurements can be reduced to a manageable level, making permanent sensor infrastructure unnecessary.**

5.2. Assumptions and Limitations

The framework was evaluated under a couple of assumptions and limitation that are worth examining.

- One key **assumption** made in this thesis is **that weather forecasts are accurate**. In the simulations, perfect forecast information is assumed, but in practice, predictions always contain uncertainty regarding the timing and amount of rainfall. In general, higher forecast uncertainty tends to reduce $IWUE$, since the controller becomes less effective at 'waiting' for expected rain events. That said, the receding-horizon nature of MPC partly compensates for this limitation. Because the optimisation is repeated daily using updated measurements and forecasts, the controller can adjust its irrigation schedule as new information becomes available. This issue is common in predictive irrigation research. For example, [9] investigated Robust MPC approaches and evaluated Certainty Equivalence Control (CE), which assumes that future disturbances, such as rainfall, are known with certainty. Their results indicate that treating deterministic forecasts as perfect can be a practical and acceptable approach to handling uncertainty in this setting. Since most predictive irrigation strategies rely on similar assumptions, the proposed framework is not more sensitive to this limitation than existing approaches.
- One limitation that should be discussed concerns how RZD measurements are obtained. In this thesis, an RZD measurement refers to measuring the soil volumetric water content (θ_i) and then calculating the RZD using Equation 2.4. As shown in that equation, the calculation also depends on the rooting depth Z_r. In the simulations, the estimated Z_r from the RAW estimator was not used when computing the RZD. Instead, the true Z_r from the simulation model was used. If the estimated Z_r had been used, the effect on the overall results and the controller's tracking performance would be minimal. The differences between the true and estimated Z_r are minor and mainly occur at the start of root development and around the moment when root growth begins to stagnate (See Figure 4.14).
- **All simulations were conducted in a noise-free environment**, which means the effect of sensor

measurement noise on the framework has not been studied. In practice, soil moisture sensors introduce noise into both the RLS parameter updates and the state estimates used by the MPC. While the RLS filter has some inherent smoothing properties that may reduce the impact of individual noisy observations, the degree to which measurement noise degrades the framework's performance remains an open question that would need to be addressed before real-world deployment.

5.3. Generalisation and Practical Implications

The framework was designed to be general, and its core components do not rely on sugarcane-specific parameters beyond what is required for any crop, such as the soil water thresholds θ_{FC} and θ_{WP}. The water balance model, Zone MPC formulation, RLS learning mechanism, and RAW estimator are all defined in a crop-independent way. That said, the evaluation was carried out only for sugarcane in Mozambique, and some characteristics of this specific setting influence how the results should be interpreted.

- **Sugarcane roots develop rapidly during the early growth phase, actively extending downward in search of water.** A consequence is that, even with relatively sparse irrigation in the early season, the expanding root zone gives the crop access to substantial soil water reserves. In the simulations, precipitation was also sufficient during this period, so very little irrigation was applied in the first 50 days, regardless of the control strategy. This makes it difficult to evaluate how well the framework handles the initial learning phase, when irrigation is actually needed from day one. The brief over-irrigation seen at the beginning of the integrated framework simulation would become more consequential in such cases. The framework's early-season performance may look better here than it would for crops with a different early growth dynamic. This observation does not necessarily undermine the framework's results, but it does motivate testing on additional crop types. For example, on crops with slower root development, or those planted in conditions where irrigation is needed immediately and continuously from the start, would provide a more demanding test to measure the initial convergence. It would also be informative to test the framework in a climate with more frequent but lower-intensity rainfall events, where the predictive benefit of MPC is exercised more continuously rather than in sporadic large events as in the Mozambique scenario.
- One practical consideration worth discussing is the effort required for measurement. The sparse measurement strategy reduces soil moisture measurements by about 65%, compared to daily measurements. For a 330-day season, this is still a non-trivial number. On the other hand, these measurements do not need to be taken by a connected sensor network; a farmer with a portable tensiometer could realistically meet this schedule. **This makes the framework potentially accessible to a smallholder who cannot afford permanent sensor infrastructure.**
- The RAW estimator, while simple in design, proved sufficient in practice. The piecewise linear approximation of root growth works well because most of the root development in sugarcane is approximately linear between measurement intervals. For crops with more irregular crop development, a more flexible estimator might be needed. However, the overall simplicity of the approach is a strength: it is interpretable, requires only a small number of Z_r measurements (around 15–17 in this study), and introduces only modest errors in the RAW reference used by the controller.
- From a broader perspective, **the central contribution of this framework is demonstrating that a reasonably performing predictive irrigation controller can be deployed at the start of a growing**

season with nothing more than soil type information and a weather station. The performance gap relative to a well-tuned static model is small. This all can have practical relevance for regions and smallholder contexts where historical datasets are unavailable or where conditions change significantly from one season to the next.

6
Conclusion

This thesis set out to design an irrigation control framework capable of performing predictive daily irrigation without relying on historical data or crop-specific models. The proposed framework, which combines Zone MPC, online model estimation through RLS, and a RAW estimator, shows that this is possible. Across five growing seasons, the framework performs comparably to a static controller preconfigured with crop-specific knowledge, even though it starts each season without prior information. The use of a sparse measurement strategy further reduces the required amount of data without noticeably degrading performance. The framework is not without limitations. Learning the system dynamics online introduces trade-offs, especially early in the season, and affects how well the controller can track seasonal changes in crop water demand. These limitations are manageable but still important, and they highlight areas where the framework could be improved. The main takeaway is that the performance gap between a tuned controller and one that learns from scratch is smaller than might be expected. For many agricultural settings where historical data is unavailable and sensor infrastructure is expensive, this is an encouraging result. It suggests that predictive irrigation control remains feasible in environments with limited resources.

6.1. Future Work

A significant improvement to the framework would likely be the addition of a dedicated online estimator for the crop coefficient K_c. As discussed in section 5.1, the RLS filter cannot reliably track the seasonal evolution of K_c when ET_o is used as input. One possible solution would be to introduce a third estimator in the framework, alongside the RLS filter and the RAW estimator, specifically designed to estimate K_c using canopy cover measurements. In [1], the authors describe how evapotranspiration can be divided into transpiration and evaporation based on the amount of green leaf area and ground cover. As the canopy develops, it intercepts more solar radiation, increasing transpiration while reducing soil evaporation by shading the ground. By leveraging this physical relationship, it would be interesting to investigate whether periodic canopy cover measurements could yield a sufficiently accurate online estimate of evapotranspiration. Such an estimator could provide a running estimate $\hat{K}_c(k)$ which, combined with ET_o from the weather station, would give $\hat{ET}_c(k) = \hat{K}_c(k) \cdot ET_o(k)$ as input to the RLS filter. This could potentially resolve the problem discussed in section 4.5. Previous work by [16] has shown that canopy cover can be an effective basis for modelling ET in sugarcane irrigation. However, their approach relies on multi-season historical datasets that are fitted offline, which does not align with the goals of the framework proposed in this thesis.

A second direction to explore is validation under real field conditions. All results in this thesis were obtained using AquaCrop as a simulation environment. While the simulator is validated across a wide range of crops and climates, it is still a model. Field experiments would expose the framework to effects that simulations cannot fully capture, such as sensor measurement noise and drift, differences between a commanded irrigation amount and what is actually applied, and the practical difficulties of manually measuring rooting depth. Field validation on crops other than sugarcane would also be valuable. As discussed in section 5.3, sugarcane's rapidly expanding early-season root system means that the consequences of the RLS filter's initial parameter uncertainty are partially masked by low irrigation demand in the first weeks.

Finally, the RAW estimator has room for improvement. The current piecewise linear approximation of rooting depth is simple and works well for sugarcane, where root development is approximately linear between measurement intervals during the active growth phase. For crops with more irregular development, this may be less accurate. A natural alternative would be to fit a sigmoid logistic function to the incoming rooting depth measurements, which would better capture the accelerating and then decelerating shape of root growth. The earlier attempt at this approach proved unstable with sparse early-season data, but further experimentation might yield different results.

References

[1] "(PDF) Model based predictive control strategy for water saving drip irrigation". en. In: *ResearchGate* (Oct. 2024). DOI: 10.1016/j.atech.2023.100179. URL: https://www.researchgate.net/publication/367094434_Model_based_predictive_control_strategy_for_water_saving_drip_irrigation (visited on 04/14/2025).

[2] *Annual freshwater withdrawals, agriculture (% of total freshwater withdrawal) | Data*. URL: https://data.worldbank.org/indicator/ER.H2O.FWAG.ZS (visited on 09/16/2025).

[3] Karl Johan Astrom and Bjorn Wittenmark. *Adaptive Control*. 2nd. USA: Addison-Wesley Longman Publishing Co., Inc., 1994. ISBN: 0-201-55866-1.

[4] Muhammad Ashar Ayaz, Talha Manzoor, and Abubakr Muhammad. "MPC Based Soil Moisture Regulation of a Canal-Connected Crop Field". en. In: *IFAC-PapersOnLine* 53.5 (2020), pp. 170–175. ISSN: 24058963. DOI: 10.1016/j.ifacol.2021.04.095. URL: https://linkinghub.elsevier.com/retrieve/pii/S2405896321001841 (visited on 03/18/2025).

[5] African Development Bank. *Republic of Mozambique*. en. Text. Publisher: African Development Bank Group. Apr. 2019. URL: https://www.afdb.org/en/countries/southern-africa/mozambique (visited on 01/26/2026).

[6] "Soil Moisture". In: *Climate Change and Terrestrial Ecosystem Modeling*. Ed. by Gordon Bonan. Cambridge: Cambridge University Press, 2019, pp. 115–133. ISBN: 978-1-107-04378-7. DOI: 10.1017/9781107339217.009. URL: https://www.cambridge.org/core/books/climate-change-and-terrestrial-ecosystem-modeling/soil-moisture/94367F1E2A95F1B991C333BB0637409C (visited on 09/24/2025).

[7] Erion Bwambale, Felix K. Abagale, and Geophrey K. Anornu. "Smart irrigation monitoring and control strategies for improving water use efficiency in precision agriculture: A review". en. In: *Agricultural Water Management* 260 (Feb. 2022), p. 107324. ISSN: 03783774. DOI: 10.1016/j.agwat.2021.107324. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378377421006016 (visited on 02/10/2025).

[8] G B Caceres et al. "Economic Model Predictive Control for Smart and Sustainable Agriculture". en. In: (). URL: https://repositorio.uloyola.es/bitstream/handle/20.500.12412/4453/Economic%20Model%20Predictive%20Control%20for%20Smart%20and%20Sustainable%20Farm%20Irrigation.pdf?sequence=1&isAllowed=y.

[9] Dilini Delgoda et al. "Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AQUACROP model". en. In: *Environmental Modelling & Software* 78 (Apr. 2016), pp. 40–53. ISSN: 13648152. DOI: 10.1016/j.envsoft.2015.12.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364815215301262 (visited on 02/10/2025).

[10] Dilini Delgoda et al. "Root zone soil moisture prediction models based on system identification: Formulation of the theory and validation using field and AQUACROP data". en. In: *Agricultural Water Management* 163 (Jan. 2016), pp. 344–353. ISSN: 03783774. DOI: 10.1016/j.agwat.2015.08.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378377415300779 (visited on 02/10/2025).

[11] FAO. "Analysis of incentives and disincentives for sugar cane in Mozambique". en. In: (). URL: https://openknowledge.fao.org/server/api/core/bitstreams/46611859-a13e-46f7-8364-0773eb2f9658/content.

[12] "Feeding the future global population". en. In: *Nature Communications* 15.1 (Jan. 2024). Publisher: Nature Publishing Group, p. 222. ISSN: 2041-1723. DOI: 10.1038/s41467-023-44588-y. URL: https://www.nature.com/articles/s41467-023-44588-y (visited on 09/16/2025).

[13] *Home | AquaCrop | Food and Agriculture Organization of the United Nations*. URL: https://www.fao.org/aquacrop/en (visited on 10/16/2025).

[14] Ya Huang et al. "Evaluation of water use efficiency and optimal irrigation quantity of spring maize in Hetao Irrigation District using the Noah-MP Land Surface Model". en. In: *Agricultural Water Management* 264 (Apr. 2022), p. 107498. ISSN: 03783774. DOI: 10.1016/j.agwat.2022.107498. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378377422000452 (visited on 12/11/2025).

[15] Md Hasibul Islam et al. "Agriculture 4.0 adoption challenges in the emerging economies: Implications for smart farming and sustainability". en. In: *Journal of Economy and Technology* 2 (Nov. 2024), pp. 278–295. ISSN: 29499488. DOI: 10.1016/j.ject.2024.09.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S2949948824000416 (visited on 11/18/2025).

[16] Ruud Kassing, Bart De Schutter, and Edo Abraham. "Optimal Control for Precision Irrigation of a Large-Scale Plantation". en. In: *Water Resources Research* 56.10 (Oct. 2020), e2019WR026989. ISSN: 0043-1397, 1944-7973. DOI: 10.1029/2019WR026989. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019WR026989 (visited on 02/12/2025).

[17] Dildar Hussain Kazmi et al. "Unpredictable Weather and Agriculture-Based Economy of Developing Countries". en. In: *Climate Change Impacts on Agriculture: Concepts, Issues and Policies for Developing Countries*. Ed. by Wajid Nasim Jatoi et al. Cham: Springer International Publishing, 2023, pp. 65–78. ISBN: 978-3-031-26692-8. DOI: 10.1007/978-3-031-26692-8_4. URL: https://doi.org/10.1007/978-3-031-26692-8_4 (visited on 09/16/2025).

[18] Lennart Ljung. "System Identification: Theory for the User, 2nd Edition". In: *IEEE Robotics & Automation Magazine* 19.2 (June 2012). ISSN: 1558-223X. DOI: 10.1109/MRA.2012.2192817. URL: https://ieeexplore.ieee.org/abstract/document/6213241 (visited on 01/19/2026).

[19] Yawen Mao et al. "Soil moisture regulation of agro-hydrological systems using zone model predictive control". en. In: *Computers and Electronics in Agriculture* 154 (Nov. 2018), pp. 239–247. ISSN: 01681699. DOI: 10.1016/j.compag.2018.09.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168169918304113 (visited on 06/25/2025).

[20] Klaas Metselaar, Everton Alves Rodrigues Pinheiro, and Quirijn De Jong Van Lier. "Mathematical Description of Rooting Profiles of Agricultural Crops and its Effect on Transpiration Prediction by a Hydrological Model". en. In: *Soil Systems* 3.3 (July 2019), p. 44. ISSN: 2571-8789. DOI: 10.3390/soilsystems3030044. URL: https://www.mdpi.com/2571-8789/3/3/44 (visited on 02/04/2026).

[21] *Mozambique climate: temperature, rain, when to go - Climates to Travel*. URL: https://www.climatestotravel.com/climate/mozambique (visited on 01/26/2026).

[22] Staff Reporter. *Sugar producers in Mozambique struggle to recover from Cyclone Freddy*. en-US. Mar. 2023. URL: https://macaonews.org/news/lusofonia/mozambique-sugar-freddy/ (visited on 01/26/2026).

[23] *The AquaCrop model – Enhancing crop water productivity*. en. FAO, Nov. 2021. ISBN: 978-92-5-135222-9. DOI: 10.4060/cb7392en. URL: http://www.fao.org/documents/card/en/c/cb7392en (visited on 12/11/2025).

[24] *Water scarcity | UNICEF*. en. URL: https://www.unicef.org/wash/water-scarcity (visited on 09/16/2025).

[25] Lincoln Zotarelli et al. "Step by Step Calculation of the Penman-Monteith Evapotranspiration (FAO-56 Method): AE459, 2/2010". en. In: *EDIS* 2010.2 (Aug. 2024). ISSN: 2576-0009. DOI: 10.32473/edis-ae459-2010. URL: https://journals.flvc.org/edis/article/view/137172 (visited on 03/10/2025).