Vol. 7(2), pp. 10-16, July-December, 2023
DOI: 10.5897/JECI2023.0206
Article Number: 160372170979
ISSN: 2141-6508
Copyright ©2023
Author(s) retain the copyright of this article
http://www.academicjournals.org/JECI

ACADEMIC JOURNALS
expand your knowledge

Journal of Engineering and Computer Innovations

Full Length Research Paper

Applying AquaCrop-OSPy to real-time irrigation

Peter G. Raeth

Creative Solutions, Beavercreek, Ohio, USA.

Received 1 March, 2023; Accepted 7 July, 2023

Crop and irrigation modeling based on fundamental physics can be very valuable in forecasting requirements and production in preparation for next season's planting/harvesting cycle. AquaCrop is such a model. It forecasts daily watering needs and allows for simple through detailed data input. Its supplied input files offer configurations for various crop, soil, terrain, and irrigation types. If one knows historical weather, it is possible to estimate daily crop water needs in the coming season, assuming the weather of the coming season is similar to previous years. AquaCrop-OSPy is an open-source implementation of AquaCrop, created in cooperation with AquaCrop's authors. Looking at this irrigation modeling tool from a computer engineering perspective, one begins to realize that AquaCrop-OSPy could be expanded to input real-time weather data and produce real-time signals to an irrigation system. A simple new console would show communication with the irrigation system. This paper discusses and illustrates a proof-of-concept for that possibility. The author started with basic software for querying up-to-date weather data and estimating ETo for a single crop under normal conditions. Those modules were coupled with a microcontroller to confirm what is possible. Those results encouraged an expansion to AquaCrop-OSPy so that it can be used to support fully-automated human-supervised irrigation. All software for this paper is available at https://github.com/SoothingMist/Embeddable-Software-for-Irrigation-Control.

Key words: Precision irrigation, precision agriculture, water conservation, AquaCrop, drought mitigation, crop forecasting, irrigation control, irrigation requirements, irrigation methods, soil moisture.

INTRODUCTION

While taking a computer engineering view of precision irrigation, the author saw opportunities for drought mitigation. Available water could be husbanded while still maintaining and possibly improving the value of the crop. Human-supervised automated irrigation supports this activity in an understandable and implementable way.

Precision irrigation applies water according to a crop's requirement, as estimated via various sensing mechanisms and ambient conditions. Good reviews of the means and challenges of precision irrigation have been published (Gundim et al., 2023; Thorp et al., 2022; Liang et al., 2020; Brahmanand and Singh, 2022; Darji et al., 2023; Awawda and Ishaq, 2023; Samreen et al., 2022; Plascak et al., 2021).

Numerous papers specific to various techniques also exist:

(1) Thermal imaging (Katz et al., 2022, 2023).
(2) Normalized Difference Vegetation Index (Chen et al., 2006; Mather, 1999; Steven, 1998).

E-mail: peter_raeth@juno.com.

(3) Direct soil-moisture measurements (Caldwell et al., 2022; Schwamback et al., 2023; Chowdhury et al., 2022).
(4) Soil moisture estimation (Allen et al., 1998; Luong et al., 2023; Sharma et al., 2018).

Regarding various means of processing associated data:

(1) Data-driven (Abolafie-Rosenzweig et al., 2019; Zhang et al., 2021; Abioye et al., 2022).
(2) Physics-driven (Allen et al., 1998; Gurmiere et al., 2020; Norizan et al., 2021).

Vast indeed is the domain of precision irrigation, as is the greater domain of precision agriculture.

In the course of this exploration, it becomes clear that a great deal of detail is needed for accurate water-need estimates to be produced for a given crop. These details involve terrain, soil, irrigation method, and daily weather conditions. Respected models already exist that take such detail into account. AquaCrop is one of these. It is drawn from basic physics (Allen et al., 1998; Martin and Gilley, 1998).

AquaCrop's executable is open-access but its source code is closed-source. AquaCrop-OSPy is a version of AquaCrop that is open-source and written in cooperation with AquaCrop's authors. Both tools forecast watering need and crop production under assumed weather conditions. It is this author's assertion that AquaCrop-OSPy could be expanded so that it uses real-time weather readings and so can act as the heart of a human-supervised automated irrigation system. This would not be a trivial modification and would require tracing through the underlying source code during its operation. This paper discusses the results of having done so. The results are available under creative-commons licensing.

Therein lies the contribution of this work. It applies to a niche within precision irrigation, extending a respected physics-based irrigation and crop-yield forecasting model, AquaCrop-OSPy, so that the model can be used not only for forecasting but for direct control of an irrigation system. This proof-of-concept is accomplished by making as little change to the model's implementation as possible so that it still uses all available input and still can be used for parameter adaptation. The difference being that real-time weather data and estimation of ETo replaces data drawn from a file. Watering need produced by the model can then be used to drive an automated human-supervised irrigation system.

This study focuses now on literature directly related to the present effort.

AquaCrop (FAO) *"is the crop growth model developed by FAO to address food security and assess the effect of the environment and management on crop production. AquaCrop simulates the yield response of herbaceous crops to water and is particularly well suited to conditions in which water is a key limiting factor in crop production."*

This is a production/irrigation forecasting tool that has not previously been aimed at real-time operation of irrigation systems. Given the amount of detail used by this model, there is much potential beyond forecasting and planning. While its executable is available open-access, its source code is not available.

Foster et al. (2016), Kelly and Foster (2021), and Kelly (2022) developed AquaCrop-OS, open source software written in M-Language that is meant to mirror the capability of FAO's AquaCrop. Their code will run under MatLab, a commercial M-Language interpreter, and Octave, an open-source M-Language interpreter. They also produced a sister version in Python (AquaCrop-OSPy). Both provide an opportunity to add real-time capability. This present project uses the Python version since that is a language that is more commonly known, in the author's experience.

Delgoda et al. (2015, 2016) used AquaCrop as a physical model to project the results of an approach to real-time irrigation control. Their work stimulates the development and testing of various control methods and acts as a means of estimating results prior to implementation.

Operating with AquaCrop-OS, Kassing et al. (2020) and Mather (1999) developed a water-use optimization method for a large farm covering various soil types and terrain. Their simulation demonstrates very encouraging results relative to actual results.

Parameter optimization of an AquaCrop-OS predictive model is carried out by Zhang et al. (2019). They present a means of predicting canopy states in real-time using Bayesian statistics. This is important to associating a model with a specific farm and crop. Their refined parameters cause their model to correspond closely to actual results.

Another success in parameter optimization is attained by Ccama et al. (2022). They use real-time data to continuously adjust AquaCrop-OS so that it provides excellent irrigation guidance.

An example of evolutionary algorithms applied to AquaCrop-OSPy parameter optimization is offered by Lyu et al. (2022). They show an optimization method over three different water climates to minimize water-use while maximizing crop yield. This is another good example of integrating AquaCrop-OSPy with external data flows and analysis techniques as part of forecasting and planning for the following year.

These papers illustrate how AquaCrop-OS can be joined with external dataflows to optimize model parameters during forecasting/planning processes. They predict optimal schedules so that water-use is minimized while irrigating farmland for best crop results. Extending beyond forecasting/planning, the author illustrates a way to apply real-time weather data so that optimized parameters resulting from forecasting/planning allow

AquaCrop-OSPy to act as the heart of a human-supervised fully-automatic irrigation system. This is a reasonable next-step given the successes of the cited work.

MATERIALS AND METHODS

The approach used to modify AquaCrop-OSPy's implementation employed the following tools:

(1) Standard AquaCrop-OSPy source code as downloaded from its public github repository.
(2) PyCharm Community, an open-access tool for working with software written in Python.
(3) Software for ETo calculations (single-crop under normal conditions derived from Allen et al.)
(4) Access to weather data via US Weather Service API.
(5) Microsoft Visual Studio, an open-access tool for working with software written in C++

Modifying AquaCrop-OSPy

Herein is described the modifications made to AquaCrop-OSPy.

Top-level design

At its most basic level, AquaCrop-OSPy begins with an initialization step. Then it initiates the simulation. Once the simulation is finished, it reports on the results. Data for the simulation comes from initialization and also from a group of files containing details on weather, crop, terrain, soil, and irrigation method. Documentation is very detailed. One can just provide pointers to existing files or develop files of one's own. Initialization can also be simple or detailed since there are defaults for everything. Thus, one can proceed according to one's knowledge and time. Figure 1 illustrates basic design and workflow.

Clearly, iterative optimizations carried out by the authors cited earlier require integration with baseline software. In the case of the present project, modifications were carried out in the initialization block and the simulation block. Underlying algorithms were not touched. Variables were only accessed, except for the weather data and the associated ETo value. That data is no longer read from a file but from an external source, as needed.

Modifications

The first modification is in the concept of operations. Originally, several years of daily historical weather data is read, all at the same time. That data becomes a table for access during the simulation. As the simulation progresses, that data is partitioned into crop seasons. Since this project deals with irrigation automation, no historical weather data is involved and only one season is assumed. Daily weather data is queried for that season. All other parameters are left as set via whatever paradigm or optimization created them. In fact, there is no reason iterative optimization could not continue, since nothing but weather data is touched.

Initialization

Initialization takes place within the AquaCrop-OSPy code module main.py. To maintain the inner workings of the simulation, the means of identifying weather data is changed:

```
Originally:
  weather_file_path = get_filepath(<climate file within the data subdirectory>)
  weather_df = prepare_weather(weather_file_path)

Now:
  weather_df = pd.DataFrame(np.array([[0.0,0.0,0.0,0.0,today_timestamp],
                                      [0.0,0.0,0.0,0.0,harvestDay_timestamp]]),
                            columns = ['MinTemp', 'MaxTemp', 'Precipitation', 'ReferenceET', 'Date'])
```

The variables today_timestamp and harvestDay_timestamp are Timestamp variables created using Python's pandas and datetime libraries. The resulting weather_df matches the internal data format as determined via code tracing. It identifies the start and end dates of the crop season. The subsequent call to AquaCropModel initializes the simulation. weather_df is required for that initialization. After initialization, the model is set to running in the standard way.

Simulation

Here is where the rest of the new programming occurs. External code modules are added so that the least disturbance is made to the existing code. Figure 2 summarizes the modifications.

External executables are simply a matter of convenience since those were already written in C++. There is nothing essential about that means of implementation.

The simulation step begins by gathering and injecting daily real-time weather data whenever a reading is called for. A query is made for all readings during the last 24 h. Those are averaged. Precipitation is summed. A vector is created that is the same as appears in the original historical weather data file.

```
Originally:
    # extract _weather data for current timestep
    weather_step = _weather_data_current_timestep(self._weather,
                                                  self._clock_struct.time_step_counter)

Now (code prepended):
    # Replaced historical weather file input with input resulting from real-time weather query.

    thisExternalRecord = DailyWeatherQueries.PerformNextQuery()
    if thisExternalRecord is None: # look at console to see error messages
      print("Unable to obtain current weather query results.")
      return

    self._weather_df.loc[self._clock_struct.time_step_counter] = thisExternalRecord
    if self._clock_struct.time_step_counter > 0:
      self._weather = np.vstack([self._weather, thisExternalRecord])
    else:
      self._weather = np.array([thisExternalRecord])

    # extract _weather data for current timestep
    weather_step = _weather_data_current_timestep(self._weather,
                                                  self._clock_struct.time_step_counter)
```

The second step occurs just prior to the daily time step checking for model termination. Code is inserted to interact with the irrigation system. Within that external module, the number of milliliters required is converted to a volume, according to the size of the field. Then the fixed-flow irrigator is triggered for an appropriate length of time.

```
# Code tracing reveals that this is the daily irrigation requirement:
# outputs.water_flux[self._clock_struct.time_step_counter, 6]
if outputs.water_flux[self._clock_struct.time_step_counter, 6] > 0.0:
   commandLine = "IrrigationSystem.exe " +\
            str(outputs.water_flux[self._clock_struct.time_step_counter, 6])
   operatingMinutes = os.system(commandLine)
   if operatingMinutes < 0: DailyWeatherQueries.secondsBeforeNextQuery = 0.0
   else: DailyWeatherQueries.secondsBeforeNextQuery =\
      DailyWeatherQueries.secondsIn24hours - (operatingMinutes * 60.0)
else:
 DailyWeatherQueries.secondsBeforeNextQuery = DailyWeatherQueries.secondsIn24hours
```

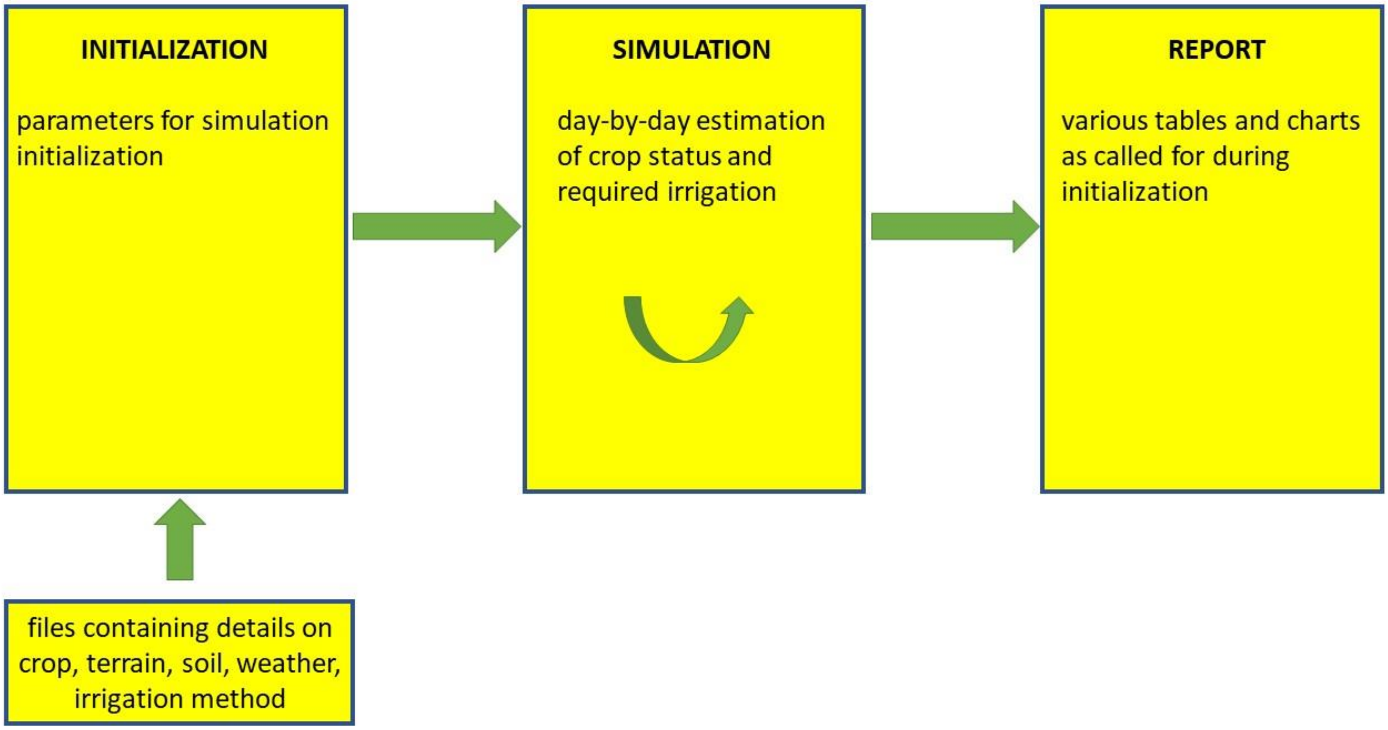
INITIALIZATION
parameters for simulation initialization
SIMULATION
day-by-day estimation of crop status and required irrigation
REPORT
various tables and charts as called for during initialization
files containing details on crop, terrain, soil, weather, irrigation method

Figure 1. AquaCrop-OSPy basic design and workflow.
Source: Author

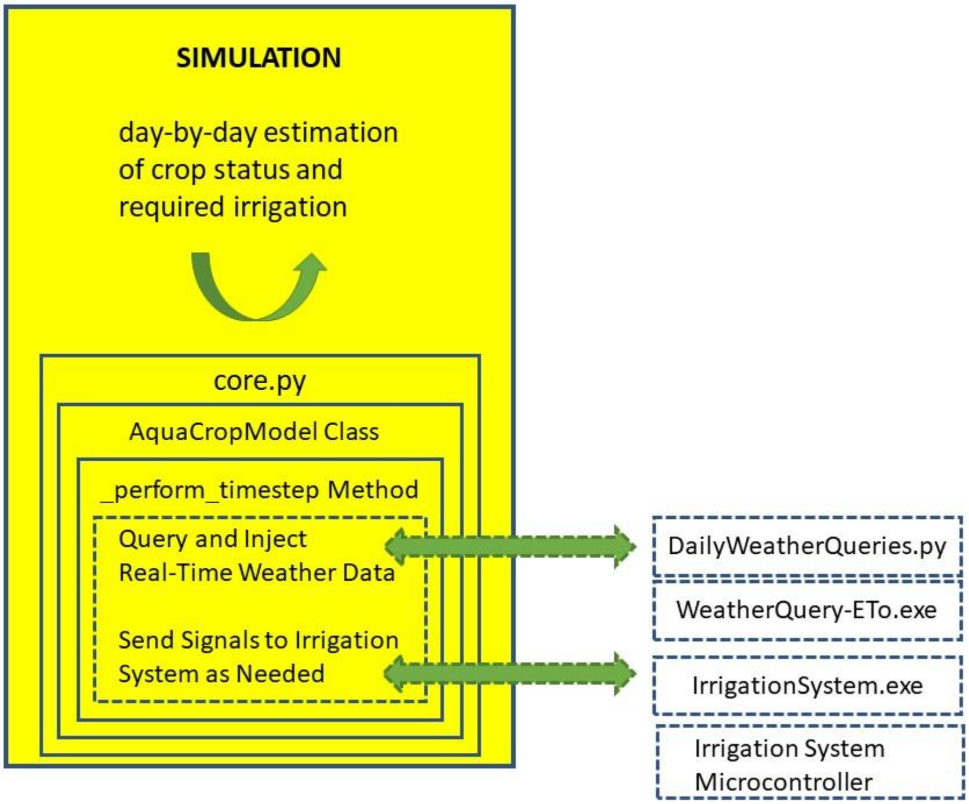
SIMULATION
day-by-day estimation of crop status and required irrigation
core.py
AquaCropModel Class
_perform_timestep Method
Query and Inject Real-Time Weather Data
Send Signals to Irrigation System as Needed
DailyWeatherQueries.py
WeatherQuery-ETo.exe
IrrigationSystem.exe
Irrigation System Microcontroller

Figure 2. Modifications to the Simulation block.
Source: Author

Abdulhameed (2016) provides a very good discussion on converting millimeters of irrigation requirement to the related volume of water relative to the size of the field. The results of this calculation are used to drive a microcontroller, which could be connected to an irrigation system. In this present project, the author assumed a constant-flow irrigation system as discussed by Netafim (2022). Therefore, the microcontroller simply stays activated for a certain length of time, according to the assumed flowrate. However, a system that does not have a constant flowrate would require the addition of a flowmeter so that the amount of water delivered can be measured. The microcontroller would turn off when the required volume has passed through the meter. This approach would be essential, for instance, in the case of an irrigation system fed by an elevated water tank.

This proof-of-concept used an Arduino Uno microcontroller development board, rather than a single-board computer (SBC) such as a RaspberryPI or BeagleBoneBlack, because the Uno has a relatively low power drain. Like SBCs, it can connect to and control external devices such a solenoids and actuators (although those were not employed in this phase, just an LED). Sensors associated with the US Weather Service were employed via internet query.

RESULTS AND DISCUSSION

Recalling Figure 2, the external executable, WeatherQuery-ETo.exe has two modes: (1) read a weather record from a file and (2) query a weather station real-time for its daily weather data. Records in the file are the exact same format and units as produced by the real-time query. Those records match the style that is normally input to unmodified AquaCrop-OSPy.

Testing was conducted in two phases. First, a single season of historic weather data was selected. AquaCrop-OSPy was then run in its pure form against that data. The output was saved for later comparison. Then modifications were made to AquaCrop-OSPy that enable external data feeds during the simulation. WeatherQuery-ETo.exe opened the same historic weather data file and fed the data to the simulation when the next day's weather was requested. The new simulation's output was compared to the original and found to be exactly the same.

After that, real-time weather data was substituted for file-based weather data. When daily weather data is needed, a query is sent to the US Weather Service for the last 24 hours of data, which is then summarized for use by the simulation. Everything proceeds as before, except that real-time weather data is employed and irrigation signals are sent. The system was run over several days to check for anomalies. None were found (There are many open-access weather services that can be queried. The author chose the US Weather Service as a matter of location, convenience, reliability, and understandability. One could also query an on-farm weather station).

The modified software was found to deliver the same results as the unmodified software. Reliability and stability of the modified software was established. AquaCrop-OSPy is already established as accurate, given its acceptance by a global community and its use in numerous forecasting studies.

Conclusion

AquaCrop-OSPy has been modified to enable real-time weather feeds and to send control signals to an irrigation system. No changes to the underlying algorithms were made. The only variable values changed are the weather-data vector. An ETo value is calculated from that vector. Everything else remains the same so that AquaCrop-OSPy operates as before, to include any iterative optimizations previously employed. In this way, we move beyond forecasting/planning to driving irrigation systems in a human-supervised automated way.

A challenge in modifying AquaCrop-OSPy was tracing through the code and finding the right variables to access, and which vector to modify for real-time weather data. A fair number of hours went into several cycles of tracing, modification, and testing. Another challenge was integrating external software modules so that the least change possible was made to the original software.

Rain prediction is another step that could be added. If rain is predicted, should irrigation take place? That is a modification that could be added to IrrigationSystem.exe (Figure 2) and would not affect AquaCrop-OSPy.

A graphical user interface would be a good step towards industrialization. At the present time, the only output provided is sent to the command console. It is not presently possible to affect the model while it is running. Still, the essential proof-of-concept has been provided.

Where AquaCrop-OSPy has been used only for forecasting, it now shows the potential for being used as the heart of a human-supervised fully-automated irrigation system. Future work could move this present proof-of-concept in that direction, while still maintaining applications in iterative parameters and forecasting/ planning. In this way, physics theory directly impacts drought mitigation.

CONFLICT OF INTERESTS

The authors have not declared any conflict of interests.

REFERENCES

Abdulhameed I (2016). How to know exactly the volume of water needed for example 7 mm irrigation rate. Department of Irrigation and Drainage Engineering, University of Anbar, Iraq, ResearchGate question response, Available at: https://www.researchgate.net/post/How-to-know-exactly-the-volume-of-water-needed-for-example-7-mm-irrigation-rate.

Abioye EA, Hensel O, Esau TJ, Elija O, Abidin MSZ, Ayobami AS, Yerima O, Nasirahmadi A (2022). Precision irrigation management using machine learning and digital farming solutions. AgriEngineering 4(1):70-103, Available at: https://www.mdpi.com/2624-7402/4/1/6.

Abolafie-Rosenzweig R, Livneh B, Small EE, Kumar SV (2019). Soil moisture data assimilation to estimate irrigation water use. Journal of Advances in Modeling Earth Systems 11(11). Available at: https://www.researchgate.net/publication/337162040_Soil_Moisture_Data_Assimilation_to_Estimate_Irrigation_Water_Use.

Allen RG, Pereira LS, Raes D, Smith M (1998). FAO Irrigation and Drainage Paper No. 56 – Crop Evapotranspiration. United Nations Report, Available at: http://academic.uprm.edu/abe/backup2/tomas/fao%2056.pdf.

Awawda J, Ishaq I (2023). IOT smart irrigation system for precision agriculture. Proceedings: Intelligent Sustainable Systems, pp. 335-346. Available at: https://www.researchgate.net/publication/367398621_IoT_Smart_Irrigation_System_for_Precision_Agriculture

Brahmanand PS, Singh AK (2022). Precision irrigation water management - Current status, scope and challenges. Indian Journal of Fertilizers 18(4):372-380. Available at: https://www.researchgate.net/profile/Anil-Kumar-Singh/publication/360119350_Precision_Irrigation_Water_Management_-_Current_Status_Scope_and_Challenges/links/6262f6a48cb84a40ac81ed34/Precision-Irrigation-Water-Management-Current-Status-Scope-and-Challenges.

Caldwell TG, Cosh MH, Evett S, Edwards N, Hofman H, Illston BG, Meyers T, Skumanich M, Sutcliffe K (2022). In situ soil moisture sensors in undisturbed soils. Journal of Visualized Experiments. Available at: https://www.researchgate.net/publication/365585346_In_situ_Soil_Moisture_Sensors_in_Undisturbed_Soils.

Ccama IB, Santoro BF, Semino JO (2022). Model predictive control for precision irrigation of a quinoa crop. Open Chemistry, review article. Available at: https://www.degruyter.com/document/doi/10.1515/chem-2022-0264/html.

Chen P, Fedosejevs G, Tiscareno-Lopez M, Arnold JG (2006). Assessment of modis-evi, modis-ndvi and vegetation-ndvi composite data using agricultural measurements. Environmental Monitoring and Assessment 119:69-82. Available at: https://naldc.nal.usda.gov/download/28732/PDF.

Chowdhury S, Sen S, Janardhanan S (2022). Comparative analysis and calibration of low cost resistive and capacitive soil moisture sensor. Internal Report, Department of Electrical Engineering, Indian Institute of Technology, Delhi. Available at: https://www.researchgate.net/publication/364222494_Comparative_Analysis_and_Calibration_of_Low_Cost_Resistive_and_Capacitive_Soil_Moisture_Sensor.

Darji P, Desai N, Bhavsar D, Pandya H (2023). Applications of remote sensing in agriculture. International Association of Biologicals and Computational Digest 2(1):108–117, Available at: https://www.researchgate.net/publication/370762708_A_REVIEW_APPLICATIONS_OF_REMOTE_SENSING_IN_AGRICULTURE.

Delgoda D, Hector M, Syed KS, Malka NH (2015). Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AquaCrop model. Elsevier Press, Journal of Environmental Modelling and Software 78:40-53, Available at: https://www.researchgate.net/publication/281967077_Root_zone_soil_moisture_prediction_models_based_on_system_identification_Formulation_of_the_theory_and_validation_using_field_and_AQUACROP_data.

Delgoda D, Syed KS, Malano H, Halgamugeb MN (2016). Root zone soil moisture prediction models based on system identification: Formulation of the theory and validation using field and AquaCrop data. Elsevier Press. Journal of Agricultural Water Management, 163:344-353. Available at: https://www.researchgate.net/publication/281967077_Root_zone_soil_moisture_prediction_models_based_on_system_identification_Formulation_of_the_theory_and_validation_using_field_and_AQUACROP_data.

Food and Agriculture Organization of the United Nations (FAO) (2023). AquaCrop downloads and documentation. Available at: https://www.fao.org/aquacrop.

Foster T, Brozovic N, Butler AP, Neale CMU, Raes D, Steduto P, Fereres, E, Hsiao TC (2016). AquaCrop-OS: An open source version of FAO's crop water productivity model. Elsevier Press, Journal of Agricultural Water Management 181:18-22. Available at: https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1014&context=wffdocs.

Gundim A, Melo V, Coelho R, Silva J, Rocha M, Franca A, Conceicao A. (2023). Precision irrigation trends and perspectives. Rural Engineering 53(8). Available at: https://www.researchgate.net/publication/368854060_Precision_irrigation_trends_and_perspectives_a_review.

Gurmiere SJ, Camporese M, Botto A, Lanfond JA, Paniconi C, Gallichand J, Rousseau AN (2020). Machine learning vs. physics-based modeling for real-time irrigation management. Frontiers in Water, Section on Water and Hydrocomplexity, 2. Available at: https://www.frontiersin.org/journals/water/articles/10.3389/frwa.2020.00008/full.

Kassing R, Schutter B, Abraham E (2020). Optimal control for precision irrigation of a large-scale plantation. Wiley Agricultural Publications, Journal of Water Resources Research 56(10), Available at: https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2019WR026989.

Kelly TD, Foster T (2021) AquaCrop-OSPy: Bridging the gap between research and practice in crop-water modeling. Elsevier Press, Journal of Agricultural Water Management, p. 254. Available at: https://www.sciencedirect.com/science/article/pii/S0378377421002419?via%3Dihub.

Kelly TD (2022). Assessing the Value of Improved Information and Management Strategies for Optimal Irrigation Scheduling. Dissertation, Department of Mechanical, Aerospace, and Civil Engineering, University of Manchester. Available at: https://pure.manchester.ac.uk/ws/portalfiles/portal/234007118/FULL_TEXT.PDF.

Katz L, Ben-Gal A, Litaor MI, Naor A, Peres M, Avira P, Alchanatis V, Cohen Y (2022). A spatiotemporal decision support protocol based on thermal imagery for variable rate drip irrigation of a peach orchard. Irrigation Science 41(21):1-19, Available at: https://www.researchgate.net/publication/364952761_A_spatiotemporal_decision_support_protocol_based_on_thermal_imagery_for_variable_rate_drip_irrigation_of_a_peach_orchard.

Katz L, Ben-Gal A, Litaor MI, Naor A, Peeters A, Goldshtein E, Ligor G, Keisar O, Rosenfeld SM, Alchanatis V, Cohen Y (2023). How sensitive is thermal image-based orchard water status estimation to canopy extraction quality. Remote Sensing, v. 15, Available at: https://www.researchgate.net/publication/369037800_How_sensitive_in_thermal_image-based_orchard_water_status_estimation_to_canopy_extraction_quality.

Liang Z, Liu X, Xiong J, Xiao J (2020) Water allocation and integrative management of precision irrigation. Water 12(11), Available at: https://www.mdpi.com/2073-4441/12/11/3135.

Luong TT, Vorobevskii I, Kronenberg R, Jacob F, Peters A, Petzoid R, Andreae H (2023). Toward reliable model-based soil moisture estimates for forest managers. Meteorologische Zeitschrift. Available at: https://www.researchgate.net/publication/370323313_Toward_reliable_model-based_soil_moisture_estimates_for_forest_managers.

Lyu J, Jiang Y, Xu C, Liu Y, Su Z, Liu J, He J (2022). Multi-objective winter wheat irrigation strategies optimization based on coupling AquaCrop-OSPy and NSGA-III: A case study in Yangling, China. Elsevier Press, Science of the Total Environment 843(4). Available at: https://www.researchgate.net/publication/361641689_Multi-objective_winter_wheat_irrigation_strategies_optimization_based_on_coupling_AquaCrop-OSPy_and_NSGA-III_A_case_study_in_Yangling_China.

Martin DL, Gilley JR (1993). Irrigation Water Requirements. National Engineering Handbook, Part 623, Section 15, Irrigation. Available at: https://www.wcc.nrcs.usda.gov/ftpref/wntsc/waterMgt/irrigation/NEH15/ch2.pdf.

Mather PM. (1999) Computer Processing of Remotely-Sensed Images, 2nd Edition. West Sussex, England: John Wiley & Sons, p. 120, Available at: https://amzn.to/440YOBd.

Netafim (2022) Corn production using subsurface drip irrigation. System Manual, Available at: https://www.netafimusa.com/4a3ea3/

globalassets/literature-usa/corn-manual.pdf.

Norizan MS, Wayayok A, Abdullah AF, Mahadi MR, Karim YA (2021). Spatial variations in water-holding capacity as evidence of the need for precision irrigation. Water 13(16). Available at: https://www.mdpi.com/2073-4441/13/16/2208.

Plascak I, Jurisic M, Radocaj D, Vujic M, Zimmer D. (2021). An overview of precision irrigation systems used in agriculture. Technical Gazette 15(4). Available at: https://www.researchgate.net/publication/355845700_An_Overview_of_Precision_Irrigation_Systems_Used_in_Agriculture.

Samreen T, Ahmad M. Baig M, Kanwal S, Nazir M, Muntaha S (2022). Remote sensing in precision agriculture for irrigation management. Environmental Sciences 23(1). Available at: https://www.researchgate.net/publication/367621928_Remote_Sensing_in_Precision_Agriculture_for_Irrigation_Management.

Schwamback D, Persson M, Berndtsson R, Bertotto LE, Kobayashi A, Wendland E (2023). Automated low-cost soil moisture sensors: Trade-off between cost and accuracy. Sensors 23(5). Available at: https://www.researchgate.net/publication/368720737_Automated_Low-Cost_Soil_Moisture_Sensors_Trade-Off_between_Cost_and_Accuracy.

Sharma PK, Kumar D, Srivastava HS, Patel P (2018). Assessment of different methods for soil moisture estimation. Remote Sensing and GIS 9(1):57-73. Available at: https://www.researchgate.net/profile/Pavan-Sharma-3/publication/325568808_Assessment_of_Different_Methods_for_Soil_Moisture_Estimation_A_Review/links/5b165361a6fdcc31bbf5953d/Assessment-of-Different-Methods-for-Soil-Moisture-Estimation-A-Review.

Steven MD (1998). The sensitivity of the OSAVI vegetation index to observational parameters. Remote Sensing of Environment 63:49-60. Available at: https://www.researchgate.net/publication/229367131_The_Sensitivity_of_the_OSAVI_Vegetation_Index_to_Observational_Parameters.

Thorp K, Calleja S, Pauli W, Thompson A, Elshikha D (2022). Agronomic outcomes of precision irrigation management technologies with varying complexity. Transactions of the American Society of Agricultural and Biological Engineers 65(1):135-150. Available at: https://www.researchgate.net/publication/357950237_Agronomic_Outcomes_of_Precision_Irrigation_Management_Technologies_with_Varying_Complexity.

Zhang J, Guan K, Peng B, Jiang C, Zhou W, Yang Y, Pan M, Franz TE, Heeren DM, Rudnick DR, Abimbola O, Kimm H, Caylor K, Good S, Khanna M, Gates J, Cai Y (2021). Challenges and opportunities in precision irrigation decision-support systems for center pivots. Environmental Research Letters 16(5). Available at: https://iopscience.iop.org/article/10.1088/1748-9326/abe436/meta.

Zhang T, Su J, Liu C, Chen WH (2019) Integration of calibration and forcing methods for predicting timely crop states by using aquacrop-os model. 2nd UK-RAS Robotics and Autonomous Systems Conference, Loughborough, UK. Available at: https://aura.abdn.ac.uk/bitstream/handle/2164/18937/Zhang_etal_UKRAS_Integration_of_Calibration_VOR.pdf?sequence=1.