ETo based irrigation system

Hi,

I started to discuss this with a forum member [IAN], how good would it be to have the OSPI using actual Eto (Evapotranspiration data) and not relying on the current (at least not for me working) Zimmerman approach.

I started to create such a script and the approach is currently only a monitoring one. Hence the control of the sprinkler (via api calls is not there yet). This lengthy post is merely to get some ideas/feedback if people think that the approach could also work for them.

I created a script (attached below) using R [which runs on a pi] to fill in the values of the table below:

	date	history	eto	rain	eto.source	rain.source	driptime	irrigation
1	3/01/2019	past	7.6	0	bom	bom	1320	7.33
2	4/01/2019	past	8.4	0	bom	bom	660	3.67
3	5/01/2019	past	8.4	0	bom	bom	0	0
4	6/01/2019	past	5.4	7	bom	bom	1320	7.33
5	7/01/2019	past	3.3	0	bom	bom	0	0
6	8/01/2019	past	5.8	11.2	bom	bom	0	0
7	9/01/2019	past	7.2	9.4	bom	bom	0	0
8	10/01/2019	past	3.43	0	Penman.recent	bom.recent	1500	8.33
9	11/01/2019	presence	2.83	9.26	Penman.owm	owm	0	0
10	12/01/2019	future	3.04	2.95	Penman.owm	owm	0	0
11	13/01/2019	future	1.86	0.44	Penman.owm	owm	0	0
12	14/01/2019	future	1.88	4.18	Penman.owm	owm	0	0
13	15/01/2019	future	3.45	0	Penman.owm	owm	0	0

It loads available data from the Australian Bureau of Meteorology [which provides Eto for previous days] and for the days where those are not available [the most recent] and future days I use formulas to estimate Eto (those formulas Penman are using Tmax and Tmin, humidity and cloud cover to predict the Eto in the coming days. Not ideal as best is to use solar radiation.). I do also download daily rainfall data.

The convenient bit is that Eto is in mm (loss of water from my lawn in a day) and so is the rain data (mm added per day)

Then I grab from my OSPI the sprinkling times (I have a drip system, hence I called it driptime) and I convert those times into [mm of water added]. The rational is simply I have about 1000 drip holes in my hoses and each drip hole distributes 2 l of water per hour. Hence on a 100 square meter lawn I drip 2l * 1000 drips = 2000 l per hour, which equates to 20 liter per square meter or 20 mm.

Hence in principle I have now a complete water balance, as all my water losses and additions are in mm.

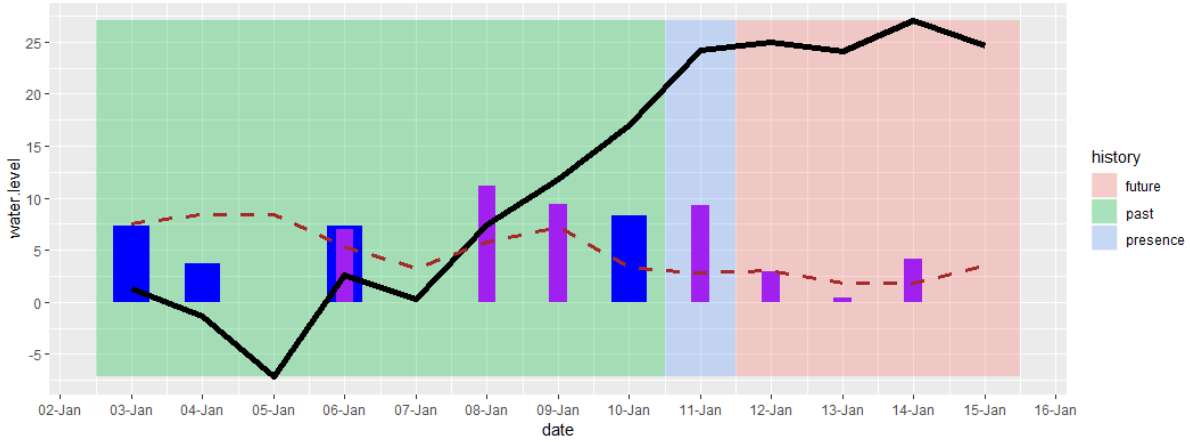
Waterlevel lawn= (rain+drip) – eto (for each day)

There are some corrections to be done (e.g. a dripping system has only ~90% effectivity and actual Eto can be converted into Etkc (the need of a certain crop type, as Eto is a reference evapotranspiration) and for lawn this means that Eto is multiplied by 0.7. E.g. I found some reference that for sports fields you do not want to “replace” 100% of the water but 70% is fine. At least that is what I found in the literature. Any comments here?

So in essence

Waterlevel = (rain +drip *0.9) – (eto * 0. 7)

For those of us that are more graphical the whole thing looks like:



black line: water level, red dotted line: ETO, blue bars: sprinkler, purple bar: rain

[All units are in mm.]

Today we have the 11th, hence I grabbed all the bom data from the previous 8 days (starting at the 3rd and used openweathermap api data to estimate Eto of today (BOM has unfortunately a delay and provides the Eto estimate only at 3 pm the following day) and also to check the predicted rainfall in my area.

As writing this the predicted rain of 9.26 mm just started, hence my water level will increase most likely if the predicted Eto of only 2.83 is correct. Hence I would most likely decide not to sprinkle in the next days.

Why do I write about this? Unfortunately the script is quite Australia specific, but if you are in Australia and apply for a free OpenWeathermap api the script should run fine and feel free to try it

(a zipped version is attached, simply add your location and apis and sprinkler password etc. at the top).

The good news would be, that anyone basically could try to create an identical table for their local conditions using whatever method and then we could aim to develop how we would like to control our sprinkler as this becomes tricky. For example it is not too hard to edit the values by hand for someone who has not those coding abilities and then the table can be used by the opensprinkler (in the best case new firmware)

As mentioned the link between the table and how the sprinkler should be controlled is the tricky bit. In terms of coding it is simple, either use api calls to do so or at some level it would be integrated in the firmware (as simple switch controlled by Eto table, preferable with some fail save tests/procedures.).

Here are my admittedly vague thoughts how such a system might work:

Aim: An efficient system (in terms of saving water) that sprinkles to have a healthy garden and to save as much water as possible.

For example, I am not 100% sure how should I determine my absolute water level. As currently I just look a certain number of days in the past (you can specify this number) and follow the relative change in the water level over days, but this is not the absolute level. One way to mitigate this would be to water the lawn to 100% at a certain day (e.g. I estimate my lawn to be "full/saturated' if 20 mm of water is applied). Then I could use those changes from the reference day onwards and I would have an absolute water level.

if the water level is then predicted (or actual) to be below say 5 mm I would water my lawn (but only if no rain is predicted in the next 24 or 48 hours. So the sprinkler would be 100% controlled from the script. This is probably a bit risky all the "errors" on the water level estimates would accumulate over time. So maybe a good idea is to reset the levels to full if there has been a lot of watering/rainfall.

Another way to program the sprinkler would be to have a fixed program that serves as a baseload and guarantees the survival of the lawn. For example every 3 days 5 mm of water is added. And the script would only moderate the number minutes if there is rain predicted or the water level falls below say 0 mm. [basically a Zimmerman approach using Eto]

Any hints how the sprinkling is managed by other systems would be highly appreciated.

Surely someone has already implemented such a system.

For all Australians I attach the script that produces the table.

Any comment is most welcome.