Modeling Rainwater Harvesting Systems with Covered Storage Tank on A Smartphone

Vikram Vyas

PhysicsInField, New Delhi 110017, India

E-mail: vikram@physicsinfield.org

Abstract: A model for simulating rainwater harvesting systems with a covered storage tank is described. The model is simple enough to serve as the simulation engine of a smartphone app for designing such systems. The results of the simulation can be used to find the optimum tank size and the reliability of the system in meeting a given demand. This is illustrated by using the model for designing two such systems, one in an urban environment and the other in an arid rural location. The Python scripts used for the simulations are included as ancillary material.

Keywords: Rainwater harvesting, water resource management, simulations

Contents

1 Introduction **1**

2 The Model **2**
2.1 Water-balance equation 2
2.2 Reliability of the RWHS 2

3 Numerical experiments: finding the optimum tank size **3**
3.1 Jaisalmer, India 4
3.2 Bangalore, India 6
3.3 Checking the approach against later rainfall 8

4 Interpretation and use of model outputs **10**
4.1 Sensitivity to the source of rainfall data 13

5 Conclusion **15**

A The SimTanka engine **16**
A.1 What the engine computes 16
A.2 A minimal example 17
A.3 Python translation and reproducibility 17

1 Introduction

Harvesting rainwater and storing it in a covered tank for future use is an important means of meeting the water needs in many parts of the world[1]. Such systems are often referred to as rainwater harvesting systems with covered storage tanks; for brevity we will refer to them simply as RWHS. These systems are attractive on many counts: they are relatively simple to build, easy to integrate with other sources of water, and they minimize the use of energy for pumping water from the source to the user. The role of RWHS is likely to grow with the increasing occurrence of drought in regions that historically have not faced water scarcity [1].

In exploring the viability of a RWHS for a given location one faces two critical questions: can it meet the given water demand reliably, and how much will it cost? The most important factor in estimating the cost of a RWHS is the size of the tank, so the second question becomes how big a storage tank will be needed. As we will see in detail later, the answers to both questions depend on the amount as well as on the variability of the rainfall, which together determine the optimum tank size, that is, the smallest tank beyond

[1] For more details see the article Rainwater Harvesting in Wikipedia.

which the reliability of the system no longer improves appreciably. It is expected that due to global warming this variability will increase in many parts of the world. This will be reflected in frequent droughts interspersed with floods [2]. Taking into account these temporal fluctuations, along with the spatial variability, requires a detailed knowledge of the rainfall pattern at the location of the RWHS [3, 4].

It is here that the advent of smartphones with GPS, and the availability of large public and commercial rainfall datasets of increasing resolution and accuracy, has opened up the possibility of developing a simple but useful model that can be converted into an app. Such an app can provide a potential user of RWHS with useful guidelines, such as the optimum tank size for a given demand, the reliability of the system, and the months when the system will be under stress and a supplemental source of water may be needed.

The problem of the optimal tank size has been much studied [4]; for examples see [5–7]. In this paper we describe and document one such model[2]; a precursor of this model was described in [3]. The outline of the paper is as follows. In Sec. 2 we describe the model. In Sec. 3 we illustrate how the model can be used to find the optimum tank size for two RWHS in India, one in Bangalore and one near Jaisalmer. The interpretation and use of the model outputs are discussed in Sec. 4, and we state our conclusions in the final section, Sec. 5. The appendix gives the details of the SimTanka engine and of its Python translation.

2 The Model

The question that we are interested in answering is how well a RWHS at a known location can meet a *given* water demand. One way of answering the question is to use a simple water balance equation to evolve the amount of water in the tank, count the number of days on which the demand cannot be met, and then increase the tank size until this number no longer decreases appreciably. For this purpose we will model the RWHS by four parameters: three that characterize the collecting surface, namely its area A, its runoff coefficient C (the fraction of the rainfall on the surface that can be collected) and its initial abstraction I_a (defined below), together with the volume V of the tank. The demand is specified by the amount of water D_i required on each day i.

2.1 Water-balance equation

Let r_i be the rainfall (mm) on day i. A small depth of rainfall is lost to wetting the surface and filling small depressions before any runoff occurs; this *initial abstraction* I_a depends on the catchment surface. The volume of water harvested on day i (m^3) is then

$$R_i = C\,A\,\max\left(0,\, r_i - I_a\right) \times 10^{-3}, \tag{2.1}$$

with A in m^2. The product CA is the *effective catchment area*. If W_{i-1} is the water in the tank at the end of the previous day and D_i the demand (m^3) on day i, we compute

$$M_i = \min\left(W_{i-1} + R_i,\, V\right) - D_i, \tag{2.2}$$

[2] The model is called SimTanka and is the simulation engine for the app RainTank. For details see simtanka.org.

Table 1. Runoff coefficient C and initial abstraction I_a for each catchment surface, as implemented in the SimTanka engine [11].

Surface	C	I_a (mm)
Metal	0.95	1.0
Plastic sheet	0.85	1.0
Concrete	0.85	1.5
Tile / shingle	0.75	2.0
Thatch	0.30	5.0
Paved (prepared) ground*	0.60	2.0
Natural sloping ground*	0.30	1.5

*Engineering assumptions; see text.

and update the tank level as

$$W_i = \max(M_i,\, 0)\,. \tag{2.3}$$

Equation (2.2) caps the stored water at the tank volume, any excess being lost as overflow, and eq. (2.3) ensures the tank cannot hold a negative amount of water. Note the order of operations: the day's harvest is added and the tank capped *before* the day's demand is drawn, so water that overflows on a given day is not available to meet that day's demand. This is a conservative choice. Given an initial level W_0, eqs. (2.1–2.3) evolve the tank level day by day.

The values of the runoff coefficient C and of the initial abstraction I_a used for different catchment surfaces are listed in Table 1. For roofs the values of I_a follow the measurements of Farreny et al. [8]; the two ground catchments, which model the prepared and the natural catchment (*agor*) of a traditional tanka [9], are not covered by those measurements, and their values are engineering assumptions taken as the midpoints of the ranges reported in standard hydrology references [10].

2.2 Reliability of the RWHS

Finally we face the question of the rainfall. The approach that we have taken is the following. Our aim is to estimate the likelihood of our RWHS meeting the given water demand. For this purpose we evolve the water in the tank using eqs. (2.1–2.3) over the daily rainfall record of the *past* five years, starting from an empty tank ($W_0 = 0$). Let N_d be the number of days on which the water demand is non-zero ($D_i > 0$), and N_s the number of those days on which the system is able to meet the demand, that is, for which $M_i \geq 0$. The reliability of the system is then reported as the percentage

$$P = \frac{N_s}{N_d} \times 100. \tag{2.4}$$

Starting with an empty tank is the cautious choice: in the first year the tank has not yet had a chance to fill, so the reliability is somewhat understated. We use it, as the app does,

to find the optimum tank. Only when we check a tank against later rainfall (Sec. 3.3) do we let it fill for one year before we start counting.

The details of these calculations, together with an explanation of the actual Swift code that implements them, are available in the *SimTanka* source repository [11]; see also Appendix A. For a limited evaluation of the future performance, the model uses the forecast of the rainfall for the next fifteen days in the water-balance equations (2.1–2.3), starting from the amount of water in the tank observed by the user, and reports the number of days on which the demand is met. In the next section we explicitly check the validity of this approach for the two sites, and we come back to the validity of its central assumption, that the recent past is a guide to the near future, in Sec. 4.

3 Numerical experiments: finding the optimum tank size

We present the results for the use of the model for determining the optimum tank size for two representative places. Each of these places is experiencing or is likely to experience water shortage. We use daily rainfall and temperature obtained from the Visual Crossing Weather API for 2012–2021, and, for the check of Sec. 3.3, for 2011–2025. Each subsection below states the catchment, roof, and demand inputs used, so the calculation is reproducible. The optimum tank size for each case is found using the engine's own convergence-based search: tank size is increased from $1\,\mathrm{m}^3$ until the reliability computed by eq. (2.4) stops improving by more than 0.05 percentage points over three consecutive steps. In each case we plot the reliability as a function of storage tank volume using both the daily rainfall for the past five years and for the past ten years, with the resulting optimum tank size for each marked on the plot. In each case we then check the validity of our approach against later rainfall, which the model has not seen (Sec. 3.3).

3.1 Jaisalmer, India

As a first example we consider a traditional *tanka*, a covered underground cistern used in the Thar desert of India, fed not by a rooftop but by a courtyard or graded ground catchment locally called an *agor* [9]. We consider a small farmer's tanka used only during the months of July and August, irrigating a small orchard of twenty ber and pomegranate trees, modelled with the engine's large-plant category. Supplemental irrigation of these fruit trees from tanka water is documented by CAZRI [12]. We consider an agor whose surface has been deliberately prepared (lime-plastered, stone, or cement-treated). Table 2 states the catchment and demand inputs used; the catchment area follows the one concrete figure available in the literature for a related structure of this kind, a *kund* [13]. The runoff coefficient C and initial abstraction I_a of the prepared surface are taken from Table 1. The crop coefficients, the equivalent areas of the plants and the irrigation efficiency used by the demand model are documented in the SimTanka repository [11]. The water demand of the orchard, computed by the demand model from the climate of each window, is shown in Fig. 1; over the two months it amounts to about $59\,\mathrm{m}^3$[3].

[3] Since the rain falling on the orchard itself is not subtracted, this demand is an overestimate, and the optimum tank errs on the safe side: in July and August the summer monsoon rain supplies much of the

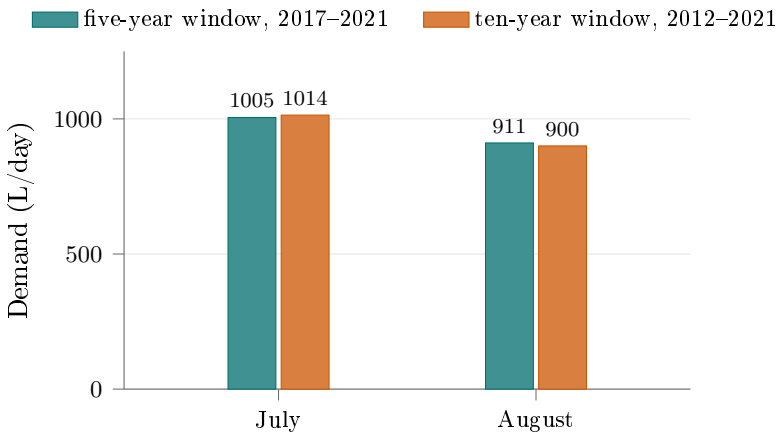

Figure 1. Daily water demand of the Jaisalmer orchard of twenty ber and pomegranate trees, in the two months in which it is irrigated from the tanka.

Table 2. Case-study inputs for Jaisalmer.

Input	Value
Latitude, longitude	26.9157°N, 70.9083°E
Elevation	225 m
Catchment area	2000 m^2
Surface, runoff coefficient	prepared ground ($C = 0.6$)
Effective catchment area	1200 m^2
Initial abstraction I_a	2.0 mm
Demand	orchard of 20 ber and pomegranate trees
	1005 / 911 L/day in July / August
Active months	July, August

The demand is computed from the reference evapotranspiration, which the engine estimates by the Hargreaves–Samani method [14] from the monthly averages of the daily temperatures, as implemented in the engine's `EvapotranspirationCalculator` (Appendix A). Since these averages are taken over the same window as the rainfall, the demand changes slightly with the window; at Jaisalmer by about 1%.

The rainfall pattern for Jaisalmer, based on the historical records for the past ten years, is shown in Fig. 2. The meagerness of the rainfall and the extreme variability is typical of the arid regions in the world. The performance of the tanka as a function of the volume of the storage tank is shown in Fig. 3. The optimum tank size is 65 m^3 for both windows, reaching 100.0% reliability for the five-year window and 92.7% for the ten-year window. This optimum is close to the total demand for July and August, about 59 m^3: the tank does not merely bridge dry days within the monsoon, but stores the season's water and carries it from one year to the next. In 2021, for example, only 11 mm of rain fell in July

trees' water, and the tanka is needed mainly in the dry spells between rains.

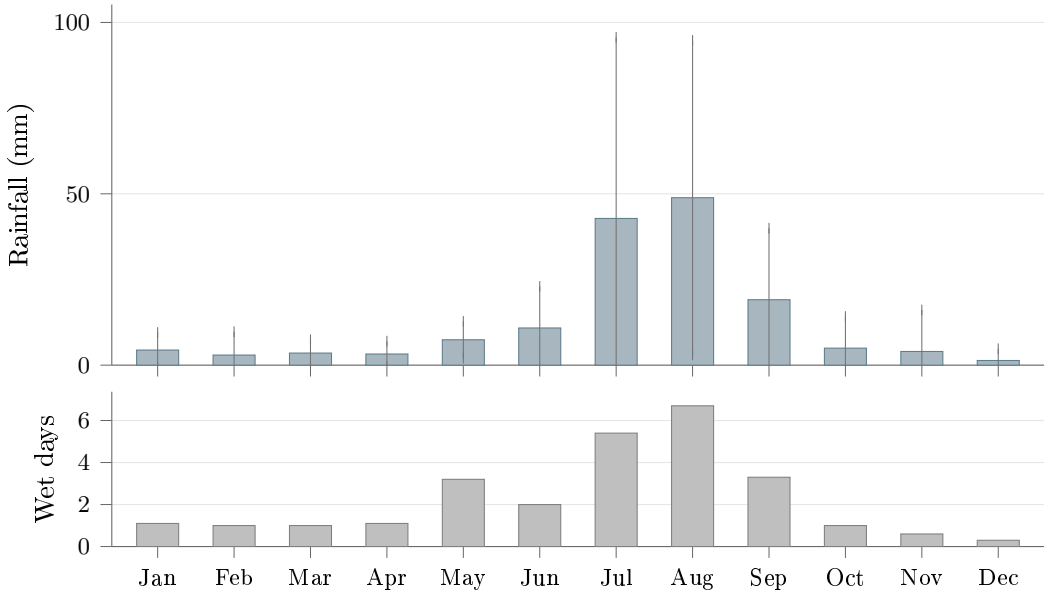

Figure 2. Monthly rainfall pattern for Jaisalmer, India, based on daily precipitation for 2012–2021. Bars show the monthly mean and whiskers show one standard deviation; the lower panel shows wet days (daily precipitation ≥ 1 mm). Data: Visual Crossing Weather API.

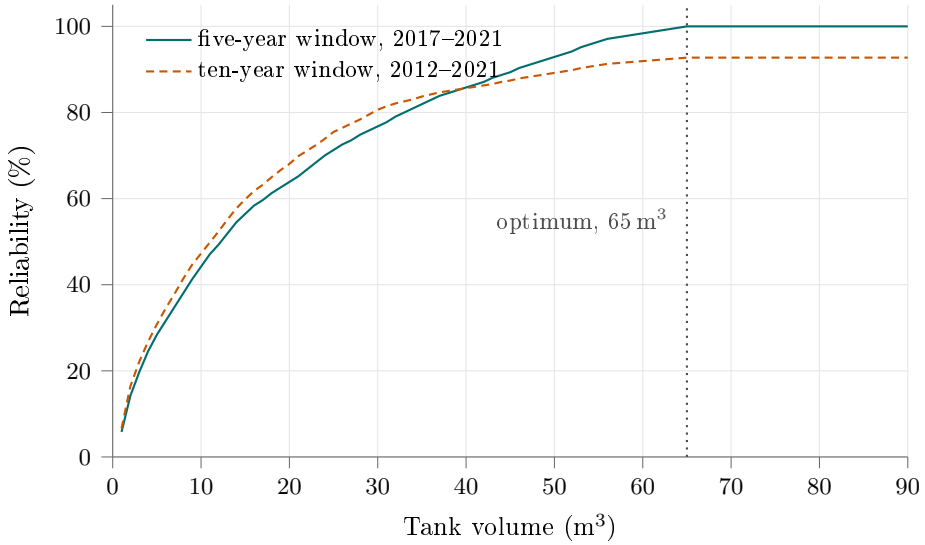

Figure 3. Performance of the Jaisalmer tanka, with a prepared agor and an orchard of twenty fruit trees, as a function of storage tank size. The dotted vertical line marks the optimum tank size, the same for both windows.

Table 3. Case-study inputs for Bangalore.

Input	Value
Latitude, longitude	12.9716°N, 77.5946°E
Elevation	920 m
Roof area, material	150 m^2, tile ($C = 0.75$)
Effective catchment area	112.5 m^2
Demand	vegetables 20 m^2,
	20 small + 10 medium + 2 large plants
Active months	December–April (dry season), or all year

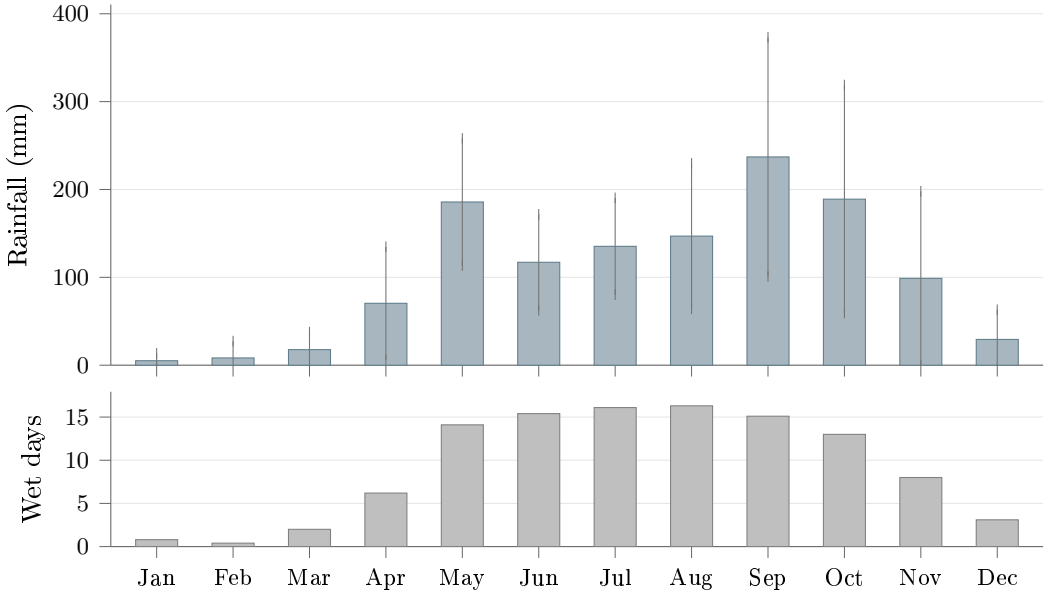

Figure 4. Monthly rainfall pattern for Bangalore, India, based on daily precipitation for 2012–2021. Bars show the monthly mean and whiskers show one standard deviation; the lower panel shows wet days (daily precipitation $\geq$ 1 mm). Data: Visual Crossing Weather API.

and August, yet the demand was met in full from water harvested in 2020.

3.2 Bangalore, India

Next we consider the case in which RWHS is used for gardening in a residential house in Bangalore, India, first only during the dry season and then throughout the year. Table 3 states the catchment, roof, and demand inputs used for this case.

The rainfall pattern for Bangalore is shown in Fig. 4. We take the dry season to be December to April, the months of little rain, for every window. The demand of the garden in each month is shown in Fig. 5. The small difference between the windows comes from the temperatures (Sec. 3.1): the ten-year window 2012–2021 was on average slightly warmer than the five-year window 2017–2021, so that the demand is slightly higher. The performance of the system is shown in Fig. 6. For the dry season the optimum tank size

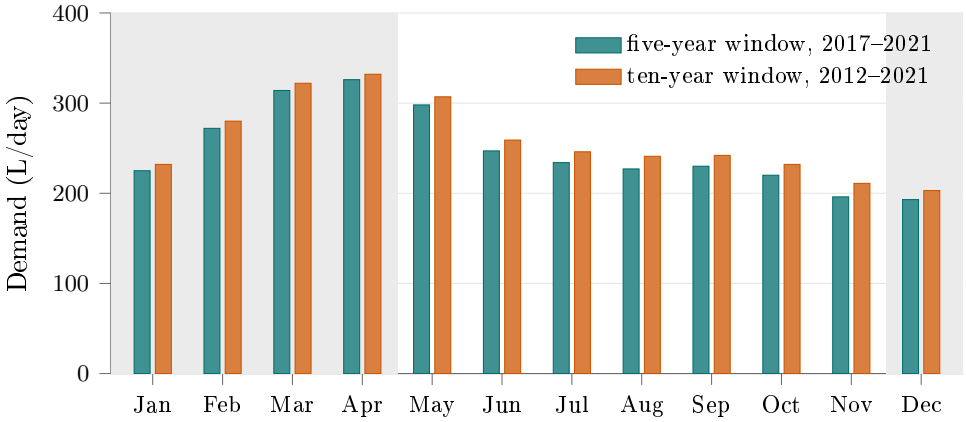

Figure 5. Daily demand of the Bangalore garden of Table 3 in each month. The shaded months, December to April, form the dry season.

is 34 m^3 for the five-year window, reaching a reliability of 86.6%, and 40 m^3 for the ten-year window, reaching 92.4%. As at Jaisalmer, the optimum is close to the demand of the season, about 40 m^3: the tank has to carry water from the monsoon into the dry months.

When the same garden is watered throughout the year, the demand is about 91 m^3 a year for the five-year window and 94 m^3 for the ten-year window, more than twice that of the dry season, yet the optimum tank is only 40 m^3, reaching 93.6%, and 56 m^3, reaching 94.5%. During the monsoon the tank is refilled as it is used, so that it has to store little more than the water of the dry season. Over the ten-year window the reliability approaches its plateau very slowly: it gains less than one percentage point between 46 and 56 m^3, so that the optimum moves a long way for little change in reliability.

When the garden is watered all year, the optimum tank of 40 m^3 meets the demand on 93.6% of the days over the five years. Figure 7 breaks this down by month. All the days on which the demand was not met fall between January and May 2017, the first year of the record, which begins with an empty tank; from 2018 to 2021 the demand was met on every day. The same holds for the dry-season garden with its optimum tank of 34 m^3.

In all cases, as expected, there is a value of storage tank size beyond which the performance remains constant; Table 4 summarizes the optimum tank size and the reliability found for each. The optimum tank size itself is the same for the five-year and ten-year windows at Jaisalmer, and larger by 18% for the ten-year window for the dry-season garden at Bangalore (40% when it is watered all year, for the reason given above). The reliability reached at the plateau differs more between the windows, mainly because the first year, which starts with an empty tank, counts for a fifth of a five-year window but only a tenth of a ten-year one.

3.3 Checking the approach against later rainfall

For each of the three systems we check the validity of our approach in the following manner. We first find the optimum tank size using the daily rainfall for 2011–2020 (ten years), then using that for 2016–2020 (five years), and finally check the performance of the system with

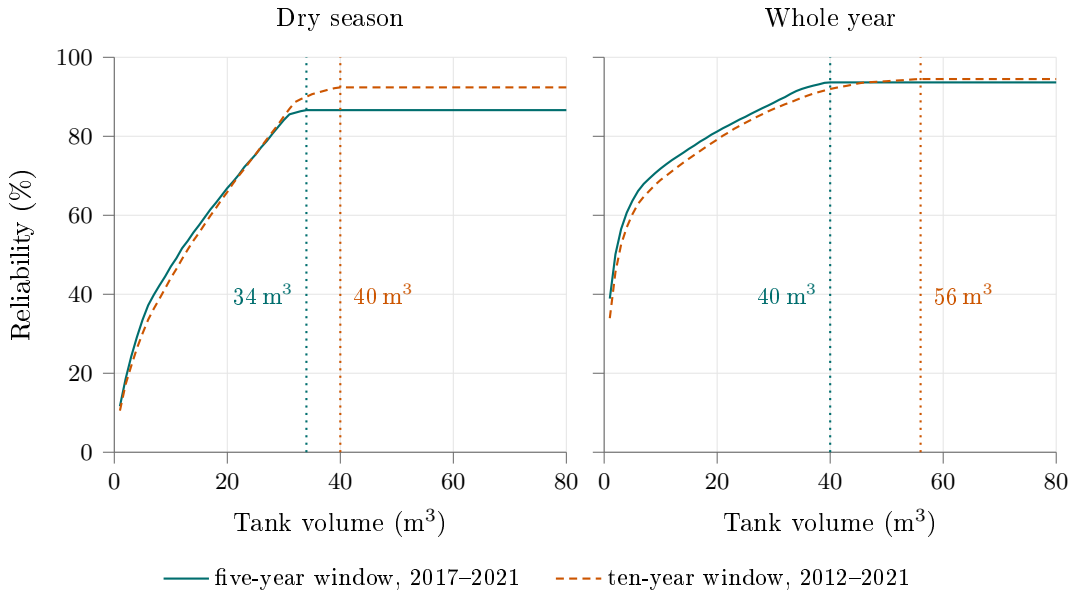

Figure 6. Performance of the Bangalore RWHS used only during the dry season, and throughout the year, as a function of the storage tank size. Dotted vertical lines mark the optimum tank size found for each window.

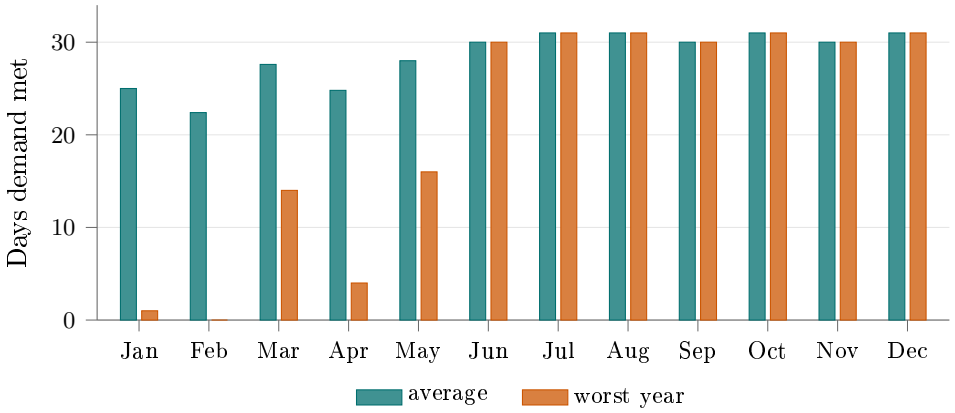

Figure 7. Monthly reliability of the Bangalore garden watered throughout the year, with the optimum tank of 40 m^3, over the five years 2017–2021: the number of days in each month on which the demand was met in full, on average and in the worst year.

Table 4. Optimum tank size and reliability at that size.

	Five-year window		Ten-year window	
Location	Tank (m^3)	Reliability (%)	Tank (m^3)	Reliability (%)
Jaisalmer	65	100.0	65	92.7
Bangalore, dry season	34	86.6	40	92.4
Bangalore, whole year	40	93.6	56	94.5

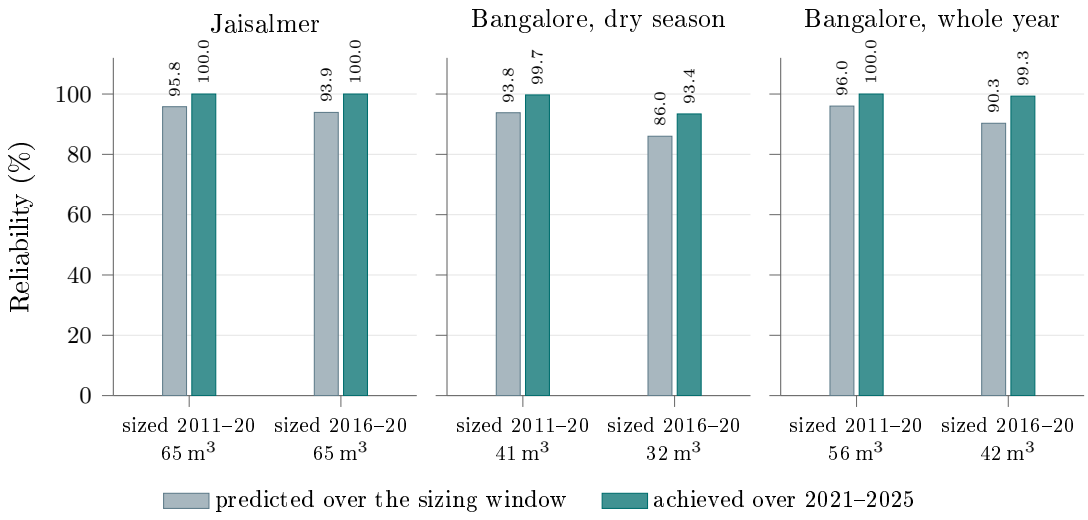

Figure 8. Check of the approach against later rainfall: for each system and sizing window, the optimum tank, the reliability predicted at that tank over the sizing window, and the reliability achieved with the rainfall of 2021–2025.

each of these tanks against the actual daily rainfall for 2021–2025, which the model has not seen. The sizing windows therefore end in 2020, one year earlier than in Secs. 3.1–3.2, so that none of the test years is used in sizing. The demand is kept at the value computed for the sizing window, since that is what a user would have planned with[4]. The predicted reliability is that of the app, which starts with an empty tank. In the test, however, the tank is not new: we let it fill during 2020 and count the days from 1 January 2021. The results are given in Figs. 8 and 9. Figure 8 gives, for each system and sizing window, the optimum tank, the reliability predicted at that tank over the sizing window, and the reliability achieved over 2021–2025; Fig. 9 gives the reliability achieved year by year.

Jaisalmer. The tank sized on either window met the demand on every day of 2021–2025, better than predicted. This includes 2021, the poorest monsoon in the record: the tank carried the water of 2020 through it.

Bangalore, dry season. The tank sized on 2011–2020 met the demand almost every day. The tank sized on 2016–2020 is smaller, 32 m^3 against 41 m^3. Over the five years it still did better than predicted, but in 2024 it met the demand on only 83% of the days (Fig. 9).

Bangalore, whole year. The tanks sized on either window met the demand on every day, or nearly so, better than predicted.

In every case the tank did at least as well as the app predicted. The app's prediction, which starts with an empty tank, is therefore on the safe side.

[4] Had the demand been recomputed from the climate of 2021–2025, it would have been 2–6% lower in every case; keeping it fixed is therefore slightly conservative.

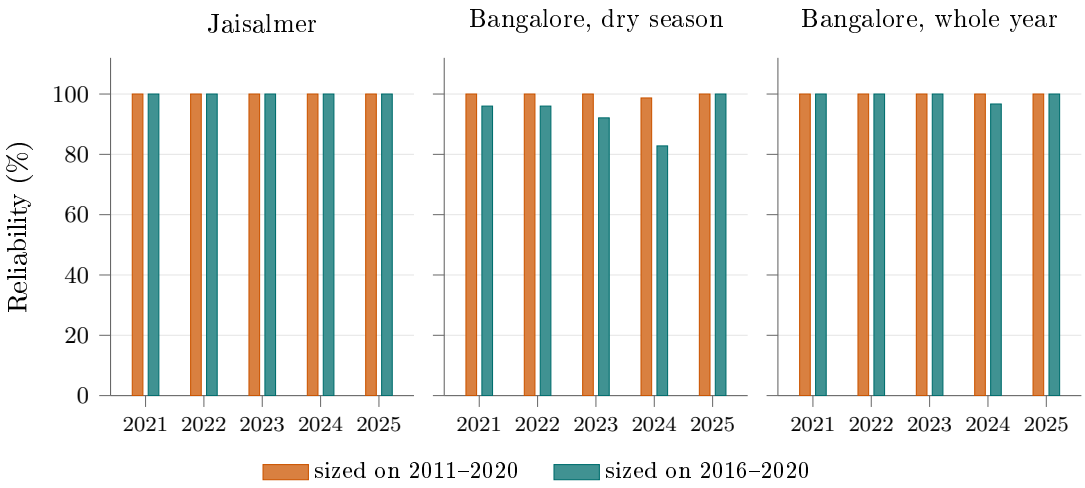

Figure 9. Reliability achieved in each year 2021–2025 by the tank sized on 2011–2020 and on 2016–2020.

4 Interpretation and use of model outputs

The main purpose for developing this model was to facilitate the design, maintenance and use of a RWHS. Some modelling choices are necessarily pragmatic. The use of the past five years of daily rainfall records to estimate future performance is an ad-hoc, but necessary, assumption in the absence of a dynamical model that can predict future rainfall with useful precision. The choice of five rather than ten years of daily rainfall records is a compromise between exposing the system to rare events and representing interannual variability. The check of Sec. 3.3 shows that, at least for the systems studied here, this is not an unreasonable approach.

We now come back to the central assumption of the model, that the recent past is a useful guide to the near future. The check of Sec. 3.3 supports it over the period 2021–2025: in every case the tank sized from the preceding five or ten years of rainfall did at least as well as predicted. Five years of test, at two sites, is a short record, and a longer record and more sites would be needed before the assumption can be regarded as established.

The model is most useful when it is used to answer two sets of practical questions:

1. Will the reliability of my system improve if I increase the existing size of the storage tank, and what reliability can I expect from the tank I can afford? What can I do to increase it: decrease the demand, or increase the size of the tank? The model answers this by computing the reliability for a range of tank sizes and finding the plateau beyond which a larger tank brings no appreciable gain, as in Figs. 3 and 6. Since the cost of the system is set mainly by the size of the tank, this is the tank worth building. Beyond the optimum size a larger tank does not help, and the reliability can then be increased only by reducing the demand; the engine finds the smallest uniform cut in demand needed to reach a target reliability. The reliability curve also gives the elasticity of the reliability, which tells how much is lost with a tank smaller

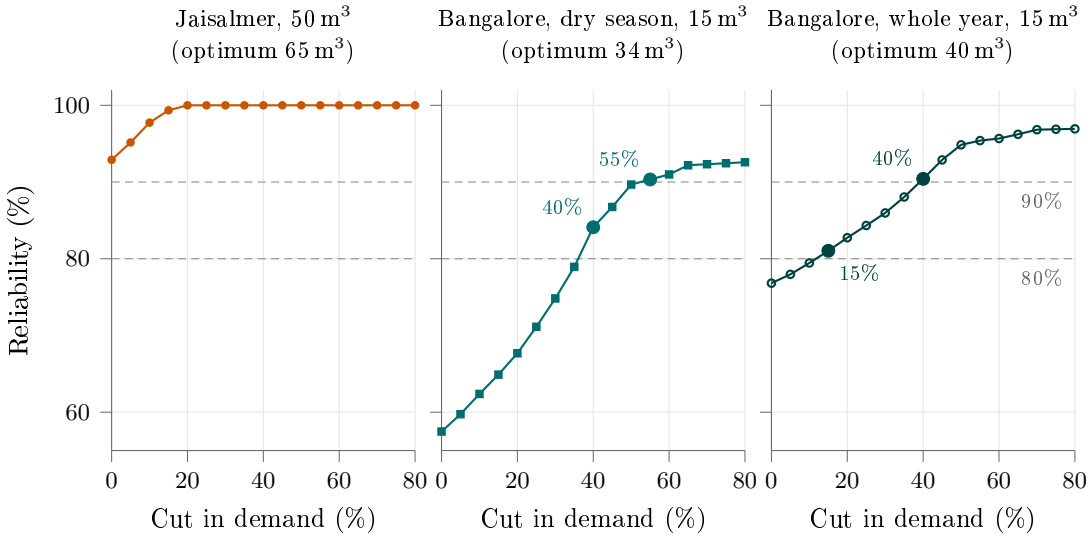

Figure 10. Reliability of tanks smaller than the optimum as the demand is reduced uniformly in every month, over the five-year window 2017–2021. Filled points mark the smallest cut that reaches 80% and 90%; at Jaisalmer no cut is needed. The demand is that of Figs. 1 and 5.

than the optimum. Examples are given below.

2. Given an initial amount of water in the tank, how might the system meet my water needs over the next fifteen days? Starting from the amount of water the user observes in the tank, the model runs the water-balance equations over the rainfall forecast and reports the days on which the demand will be met. This allows the user to plan ahead, for example by arranging a supplemental supply or by reducing use, before the tank runs dry.

To illustrate the first question, consider a user who cannot afford the optimum tank. At Jaisalmer, CAZRI recommends a tanka of 50 m^3 for a small nursery of about 200 plants [15]; built in masonry, such a tanka cost about Rs 95,000 at 2008 rates, and the cost per litre of the larger tankas changes little with size, so the optimum tank of 65 m^3 would cost roughly 30% more. In Bangalore, where rainwater harvesting has been mandatory since 2009 [16], the regulations ask for a storage of only 20 litres per square metre of roof area, that is 3 m^3 for this roof, a tank far too small to carry water from the monsoon into the dry season; the house considered here has in fact been built with a tank of 15 m^3. Figure 10 shows, for the tanks of 50 m^3 at Jaisalmer and 15 m^3 at Bangalore, watered in the dry season and all year, how the reliability over the five-year window increases as the demand is reduced uniformly in every month; the engine reports the smallest such cut, in steps of 5%, that reaches a target reliability.

At Jaisalmer the smaller tank, about a quarter cheaper, still meets the demand on 93% of the days over the five-year window, and no cut is needed to reach either 80% or 90%; over the ten-year window it reaches 89.2%, and a cut of 5% takes it to 90%. The shortfall falls in the drier years: in the worst July, in 2019, the demand was met on 16 of its 31 days.

In Bangalore the tank of 15 m^3, less than half the optimum, meets the dry-season demand on 58% of the days. A cut of 40% in the demand reaches 80%, and a cut of 55% reaches 90% (the same cuts over the ten-year window). When the garden is watered all year the same tank does better, meeting the demand on 77% of the days, and cuts of 15% and 40% reach 80% and 90% (20% and 45% over the ten-year window). The optimum found by the plateau rule is therefore not a sharp requirement. Where the reliability approaches its plateau gently, as at Jaisalmer or for the garden watered all year, a tank well below the optimum loses little; where it rises steeply up to the plateau, as for the dry-season garden at Bangalore, it loses much more. We return to this below (Fig. 11 and eq. (4.1)).

Finally, the uniform cut is only a gross indication of the quantum of reduction needed; how the demand is actually reduced, and in which months, can be decided only by the user.

Why do the two sites respond so differently to a tank smaller than the optimum? To see the shape of the reliability curve itself, we let the tank fill during 2011, count the days of 2012–2021, and scale the tank by its optimum. Figure 11 shows the reliability so obtained for the three systems. A single number characterises each curve, the elasticity of reliability at half the optimum tank,

$$E(0.5) = \frac{R(V_{\mathrm{opt}}) - R(0.5\,V_{\mathrm{opt}})}{0.5\,R(V_{\mathrm{opt}})}, \tag{4.1}$$

the fractional loss of reliability per fractional reduction of the tank over the second half of the tank, which is the slope of the dashed lines in Fig. 11. It is 0.25 at Jaisalmer and 0.61 for the dry-season garden at Bangalore, but only 0.23 when the same garden is watered all year. What decides the sensitivity is therefore when the water is used rather than the site. When the garden is watered through the rainy months, the rain refills the tank almost as fast as the water is used, so even a small tank meets the demand on most days; a larger tank helps only in the dry spells. A smaller tank therefore costs little reliability. When the demand falls in a season with almost no rain the whole season's water must be stored in advance, and the reliability grows almost in proportion to the tank.

The elasticity tells the user how much the optimum matters. Where it is high, as for the dry-season garden, the optimum is close to the tank actually needed. Where it is low, as at Jaisalmer or for the garden watered all year, the optimum is only an upper limit, and a much smaller tank loses little reliability. The final choice is therefore always the user's. In the app, the user is shown the optimum tank and its reliability, and is asked to enter the size of the tank she has or plans to build; its reliability is displayed at once. The user can thus try different sizes and balance reliability against cost in an informed way.

The results of Sec. 3 also raise the question of how stable the optimum tank is between the five-year and ten-year windows.

Jaisalmer. The optimum tank is 65 m^3 for both windows.

Bangalore, dry season. The optimum tank changes from 34 to 40 m^3 between the windows.

Bangalore, whole year. The optimum tank moves from 40 to 56 m^3, but over the ten-year window the reliability rises by less than one percentage point beyond 46 m^3. Near its

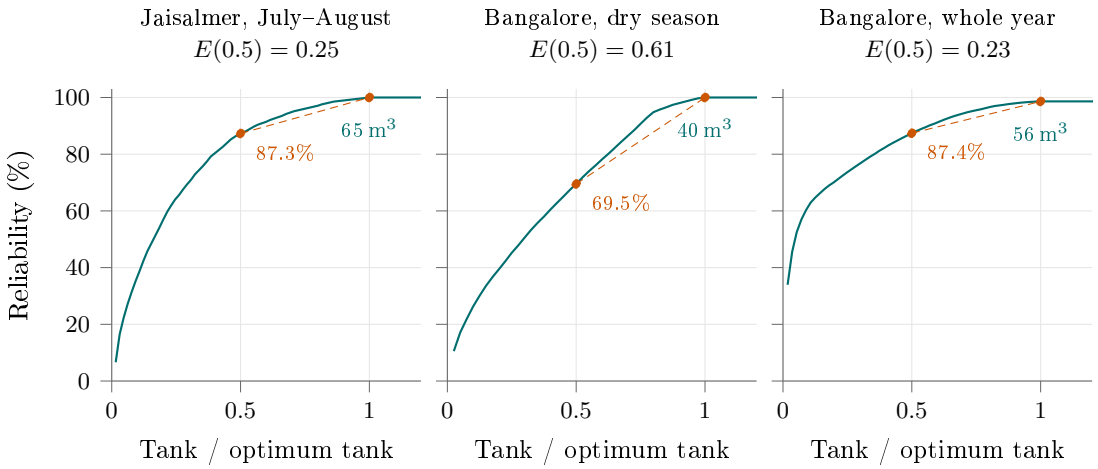

Figure 11. Reliability against the tank scaled by its optimum, for 2012–2021 with 2011 as the warm-up year. The dashed line joins half the optimum tank to the optimum; its slope is $E(0.5)$. The optimum tank is marked in each panel.

plateau this curve is so flat that the exact optimum matters little, as its small elasticity $E(0.5)$ also shows.

Whether the optimum stays as stable over longer records and at other sites is worth exploring.

4.1 Sensitivity to the source of rainfall data

The results of this paper were computed with daily data from Visual Crossing. To see how much they depend on the source, the whole analysis was repeated, with the same scripts and the same dry season, on the ERA5 reanalysis [22] for the same coordinates, 2011–2025, as served by Open-Meteo [23][5]. Figure 12 compares the optimum tanks.

Jaisalmer. The two records are almost the same, day by day, so the results hardly differ. The optimum tank is 65 m^3 with Visual Crossing and 62 m^3 with ERA5 (Fig. 12).

Bangalore. Here the two records are quite different. ERA5 has about a fifth less rain than Visual Crossing, spread over more days. This is because ERA5 does not record the rain at one place: it estimates the average rainfall over an area some 30 km across [22], and averaging over an area spreads the rain more thinly over more days.

Bangalore, dry season. The optimum tank found from the two records differs by at most 2 m^3 (Fig. 12). In the check of Sec. 3.3, the tank sized on the five years 2016–2020 is 32 m^3 with Visual Crossing data and 36 m^3 with ERA5 data. Does the result of the check depend on the tank, or on the rainfall used to test it? To find out, we tested both tanks against the rainfall of both records (Fig. 13). Each tank did about as well with either

[5] By default Open-Meteo serves a "best match" that combines ERA5 with the ECMWF operational forecast model from 2017 onwards, so that its record is not consistent in time; with that record the optimum tank at Jaisalmer falls from 65 to about 50 m^3. The ERA5 record alone is obtained with the option `models=era5`, which the scripts provide. So that the results of this section can be reproduced exactly, and without any key, the ERA5 data used here are included in the ancillary files, in the folder `data-era5`.

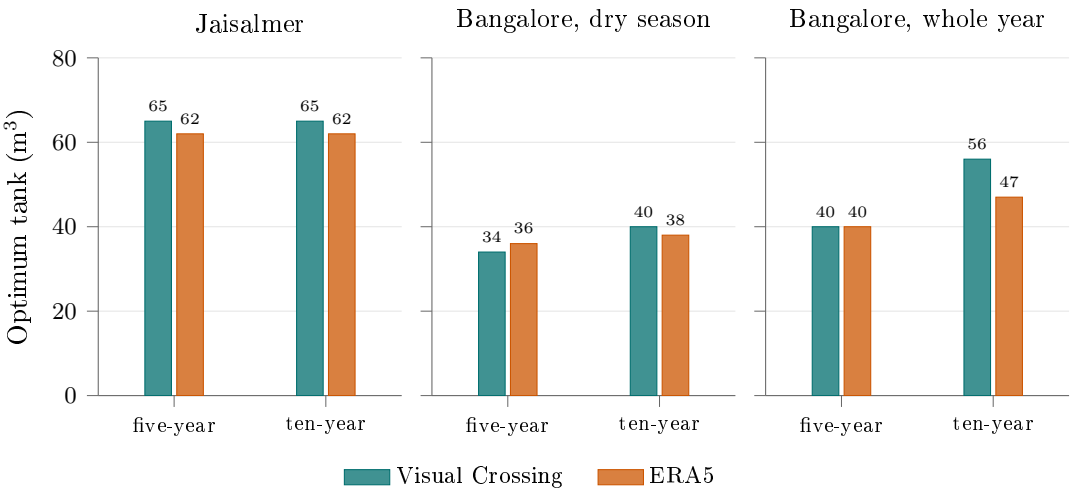

Figure 12. Optimum tank for the five-year and ten-year windows, found with the two sources of daily data.

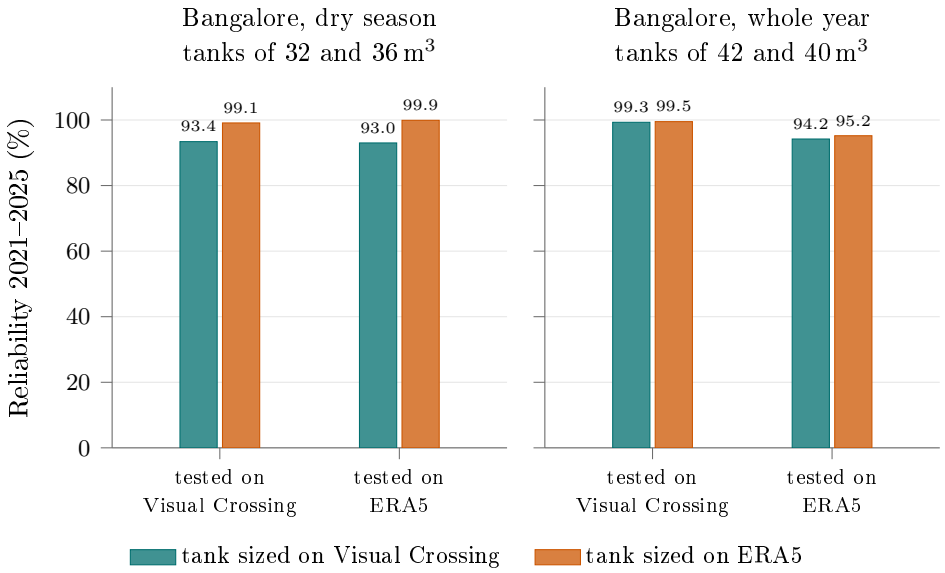

Figure 13. The tanks sized on 2016–2020 from each record, tested against the rainfall of 2021–2025 from each record, for the dry-season garden and the garden watered all year.

rainfall, and both did better than predicted; the larger tank met the demand on more days. So it is the size of the tank that matters, not the record used to test it.

Bangalore, whole year. Here the reliability rises only slowly as the tank grows beyond about 40 m^3, so a small difference in the rainfall can move the optimum a long way. For the ten-year window it is 56 m^3 with Visual Crossing and 47 m^3 with ERA5 (Fig. 12). In the check of Sec. 3.3, the two tanks performed about equally well (Fig. 13).

In short, the two records give approximately the same tank, and that is what matters to the user. We comment further on this in the Conclusion (Sec. 5).

5 Conclusion

RWHS are useful means of meeting the water needs in the water scarce regions of the world. Their attractiveness is tempered by their potential unreliability as they depend on the vagaries of the daily rainfall. The aim of the model described here is to provide a practical guideline for designing a RWHS, in terms of the optimal tank size and expected reliability. For the examples studied here, the model gives a clear answer to how large the tank should be. Just as important, it gives the reliability of any other tank, so that the user can weigh reliability against cost. When these tanks were tested against the rainfall of 2021–2025, which was not used to size them, they did at least as well as predicted. Two global datasets that differ a good deal at a given place also give approximately the same tank size (Sec. 4.1).

A natural question is whether the probabilistic model described here can be replaced, or supplemented, by machine learning models. Although earlier assessments of machine learning approaches to rainfall prediction were rather cautious [17], recent advances in deep learning and AI-based weather forecasting have substantially improved their capabilities [18]. Nevertheless, predicting daily rainfall at a specific location, particularly over the extended periods relevant to rainwater harvesting systems, remains a significant challenge. A promising possibility is to bring together local rainfall measurements and a real-time record of the water level in the tank. From these, the parameters of the model, such as the runoff coefficient and the demand, can be estimated for the actual system, and AI-based methods could use the same data to make better short-term predictions of how the system will perform.

In this context, the development of inexpensive automated rain gauges and tank water-level sensors, capable of communicating with smartphones through open protocols, would be of considerable benefit. Recent developments in low-cost wireless hydrological monitoring systems demonstrate the technical feasibility of such an approach [19], although questions of calibration, reliability, and long-term maintenance remain important [20].

Box and Draper remarked that "essentially, all models are wrong, but some are useful" [21]. A real RWHS can never be described fully by a model with only four parameters. The real question, then, is: is this model useful? SimTanka was developed to help a user balance two conflicting aims: to catch as much rain as possible, reliably, and to keep the cost low. The results of this paper indicate that, to this extent, the answer is yes.

Use of AI

The major use of AI was in translating the SimTanka engine from pure Swift into the Python scripts that are included as ancillary material. The appendix describing the engine was written by an AI model under my supervision. In addition, AI was used to improve the figures and tables, to find relevant references, and in preparing the LaTeX document. I take full responsibility for the content of the manuscript.

Acknowledgments

The development of SimTanka was started when I was working with the Ajit Foundation, and I would like to thank the Foundation for its support and encouragement. I am grateful to Janaki Abraham for her continuing support for this project, and to Visual Crossing Weather for generously providing the historical rainfall data used to develop and refine the SimTanka model. I also thank Open-Meteo for making its historical weather data freely available; the data used in Sec. 4.1 are from Open-Meteo.com (CC BY 4.0) and contain modified Copernicus Climate Change Service information.

A The SimTanka engine

The model of Sec. 2 is implemented in *SimTanka*, an open-source engine written in Swift and released under the MIT licence [11]. It is distributed as a Swift package (`RainTankEngine`) that depends only on the Foundation library and builds on macOS, iOS and Linux. The engine is purely a computation library: every function takes plain values as arguments and returns plain values, with no networking, storage or user interface. In particular it fetches no weather data. The user supplies the daily rainfall and, for the demand model, daily temperatures, latitude and elevation, from whatever source they choose. Each equation and constant is documented, with its source, in the repository, so that the model can be reimplemented in other languages.

A.1 What the engine computes

- *Historical reliability*: P of eq. (2.4) over the most recent five years of rainfall.
- *Optimum tank size*: the plateau search of Sec. 3.
- *Monthly reliability*: the same continuous simulation, broken down by calendar month and reporting both the average and the worst year, which shows *when* in the year the system tends to fail.
- *Uniform demand reduction*: the smallest uniform cut in demand, searched in steps of 5% up to 80%, needed to reach a target reliability with a given tank.
- *Forecast*: the fifteen-day projection of Sec. 2.2.
- *Water demand*: the monthly demand profile from household and garden sub-models, documented in the repository.

The illustrations in Sec. 3 use the first three.

A.2 A minimal example

The following computes the reliability of a system and its optimum tank size. The catchment is described by its effective area CA and initial abstraction (Table 1), the demand by twelve monthly values in litres per day, and the rainfall by daily records keyed by year.

```
import RainTankEngine

let surface = RoofMaterial.tile
```

```
let config = SystemConfiguration(
    effectiveCatchmentArea: 40.0 * surface.runoffCoefficient,
    tankVolume: 5.0,                        // m^3
    monthlyDemands: demandLitresPerDay,     // 12 values
    initialAbstractionMM: surface.initialAbstractionMM)

let engine = SimTankaEngine()
let result = try engine.calculateHistoricalReliability(
    config: config, rainfallData: rainfall)
let optimum = try engine.findOptimalTankSize(
    config: config, targetReliability: 1.5, rainfallData: rainfall)
```

A target above 1 (that is, above 100%) can never be reached, so the search stops only when the reliability curve levels off, as in Sec. 3.

A.3 Python translation and reproducibility

For this paper the engine and the demand model were translated line by line into Python. The one deliberate change is that the length of the window is a parameter rather than fixed at five years. This applies both to the rainfall used in the simulation and to the climate statistics from which the garden demand is computed, so a ten-year run differs from a five-year run in its demand as well as in its rainfall. The translation was checked against the Swift engine by running both on identical inputs, for the five-year window 2017–2021 (the one fixed in the Swift implementation) and for the systems of this paper: the Jaisalmer orchard, the Bangalore garden watered in the dry season, and the same garden watered all year. For all three, the reliabilities agree exactly at every tank size from 1 to 100 m^3, the optimum tank sizes are identical, the monthly reliability and the uniform cut in demand agree exactly, and the monthly demands agree to within floating-point rounding (relative differences below 10^{-14}). The check against later rainfall (Sec. 3.3) and the elasticity of Fig. 11 use the same translated water balance with a warm-up year, which is not part of the Swift engine.

The ancillary files of this paper contain the Python translation of the engine and of the demand model (files `simtanka_engine.py`, `water_demand.py` and `monthly_reliability.py`), the definitions of the two systems (`cases.py`), a single script (`paper_results.py`) that reproduces every figure and table of Secs. 3 and 4 from the daily weather, a module that reads and checks the weather files (`rainfall_input.py`), and the script used to download the weather data (`fetch_data.py`). The same files are in the folder `paper/arxiv-v2.2` of release v2.3 of the SimTanka repository [11], together with the agreement check described above (folder `parity`). The scripts read the daily rainfall (and, for the demand model, temperature) from a simple text file in a standard format, so that any source can be used, including a local rain gauge. Such records are freely available from public sources, and the included module downloads them directly, without any key, from the Open-Meteo historical weather service [23] and from NASA POWER [24], both of which provide daily rainfall and temperature worldwide; from Open-Meteo it can fetch the ERA5 record alone, as used in

Sec. 4.1, and the script `cross_test.py` repeats the cross-check of that section. Other free sources are CHIRPS [25], which provides daily rainfall between latitudes 50°S and 50°N, and, for India, the gridded daily rainfall and temperature data of the India Meteorological Department [26]. The data used in this paper, for 2011–2025, were obtained from the Visual Crossing Timeline API, whose licence does not allow them to be distributed.

References

[1] Wehner MF, Arnold JR, Knutson T, Kunkel KE, LeGrande AN. Droughts, floods, and wildfire. In: Wuebbles DJ, Fahey DW, Hibbard KA, Dokken DJ, Stewart BC, Maycock TK, editors. Climate Science Special Report: Fourth National Climate Assessment. Washington, DC, USA: U.S. Global Change Research Program; 2017. p. 231–256. Available from: https://repository.library.noaa.gov/view/noaa/32383/noaa_32383_DS1.pdf.

[2] Pendergrass AG, Knutti R, Lehner F, Deser C, Sanderson BM. Precipitation variability increases in a warmer climate. Scientific Reports. 2017;7(1):17966. Available from: https://doi.org/10.1038/s41598-017-17966-y.

[3] Vyas V. A Model for Simulating Rainwater Harvesting Systems with Closed Tanks. Ajit Foundation; 1995. Available from: https://www.ircwash.org/sites/default/files/213.2-95MO-14785.pdf.

[4] Semaan M, Day SD, Garvin M, Ramakrishnan N, Pearce A. Optimal sizing of rainwater harvesting systems for domestic water usages: a systematic literature review. Resources, Conservation & Recycling: X. 2020;6:100033. Available from: https://doi.org/10.1016/j.rcrx.2020.100033.

[5] Liaw CH, Tsai YL. Optimum storage volume of rooftop rain water harvesting systems for domestic use. Journal of the American Water Resources Association. 2004;40(4). Available from: https://doi.org/10.1111/j.1752-1688.2004.tb01054.x.

[6] Campisano A, Modica C. Optimal sizing of storage tanks for domestic rainwater harvesting in Sicily. Resources, Conservation and Recycling. 2012;63:9–16. Available from: https://doi.org/10.1016/j.resconrec.2012.03.007.

[7] Fewkes A. The verification of a behavioural model for simulating the hydraulic performance of rainwater harvesting systems. In: Memon FA, Ward S, editors. Alternative Water Supply Systems. London: IWA Publishing; 2014. p. 47–62.

[8] Farreny R, Gabarrell X, Rieradevall J. Roof selection for rainwater harvesting: Quantity and quality assessments in Spain. Water Research. 2011;45(10):3245–3254. Available from: https://doi.org/10.1016/j.watres.2011.03.036.

[9] Vangani NS, Sharma KD, Chatterji PC. Tanka: A Reliable System of Rainwater Harvesting in the Indian Desert. CAZRI Publication No. 33. Jodhpur, India: Central Arid Zone Research Institute; 1988. p. 1–24.

[10] Viessman W Jr, Lewis GL. Introduction to Hydrology. 5th ed. New York: Pearson Education; 2003.

[11] Vyas V. SimTanka: an open-source rainwater harvesting simulation engine and water demand model. Codeberg repository; 2026. Available from: https://codeberg.org/vikram_physicsInField/SimTanka.

[12] Faroda AS, Singh M, editors. Fifty Years of Arid Zone Research in India. Jodhpur, India: Central Arid Zone Research Institute; 1998. p. 4.

[13] Kunds of the Thar Desert. Centre for Science and Environment. Available from: http://www.rainwaterharvesting.org/methods/traditional/kunds.htm.

[14] Hargreaves GH, Samani ZA. Reference crop evapotranspiration from temperature. Applied Engineering in Agriculture. 1985;1(2):96–99. Available from: https://doi.org/10.13031/2013.26773.

[15] Goyal RK, Issac VC. Rainwater Harvesting through Tanka in Hot Arid Zone of India. Jodhpur, India: Central Arid Zone Research Institute; 2009.

[16] The Bangalore Water Supply and Sewerage (Amendment) Act, 2009, Section 72A, and the Bangalore Water Supply and Sewerage Board guidelines for rainwater harvesting. Available from: https://www.kscst.org.in/rwh_files/pdf/BWSSB_RWH_Act.pdf.

[17] Hussein EA, Ghaziasgar M, Thron C, Vaccari M, Jafta Y. Rainfall Prediction Using Machine Learning Models: Literature Survey. Artificial Intelligence for Data Science in Theory and Practice. 2022:75-108. Available from: https://www.researchgate.net/publication/359750167_Rainfall_Prediction_Using_Machine_Learning_Models_Literature_Survey.

[18] Lam R, Sanchez-Gonzalez A, Willson M, Wirnsberger P, Fortunato M, Alet F, et al. Learning skillful medium-range global weather forecasting. Science. 2023;382(6677):1416–1421. Available from: https://doi.org/10.1126/science.adi2336.

[19] Abdelal Q, Al-Hmoud A. Low-cost, low-energy, wireless hydrological monitoring platform: design, deployment, and evaluation. Journal of Sensors. 2021;2021:8848955. Available from: https://doi.org/10.1155/2021/8848955.

[20] Zhu Q, Cherqui F, Bertrand-Krajewski JL. End-user perspective of low-cost sensors for urban stormwater monitoring: a review. Water Science and Technology. 2023;87(11):2648–2684. Available from: https://doi.org/10.2166/wst.2023.142.

[21] Box GEP, Draper NR. Empirical Model-Building and Response Surfaces. New York: Wiley; 1987. p. 424.

[22] Hersbach H, Bell B, Berrisford P, et al. The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society. 2020;146(730):1999–2049. Available from: https://doi.org/10.1002/qj.3803.

[23] Open-Meteo. Historical Weather API. Available from: https://open-meteo.com/en/docs/historical-weather-api.

[24] NASA Langley Research Center. POWER: Prediction Of Worldwide Energy Resources. Available from: https://power.larc.nasa.gov.

[25] Funk C, Peterson P, Landsfeld M, Pedreros D, Verdin J, Shukla S, et al. The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. Scientific Data. 2015;2:150066. Available from: https://doi.org/10.1038/sdata.2015.66.

[26] Pai DS, Sridhar L, Rajeevan M, Sreejith OP, Satbhai NS, Mukhopadhyay B. Development of a new high spatial resolution ($0.25^\circ \times 0.25^\circ$) long period (1901–2010) daily gridded rainfall data set over India and its comparison with existing data sets over the region. Mausam. 2014;65(1):1–18.