

IAC-23,B2,7,12,x80188

A tracking solution via a network of beacons on the surface of Mars using the Tumbleweed mobile impactors

Elemer San Miguel^{a*},

^aTeam Tumbleweed, Delft, The Netherlands

* Corresponding Author, Elemer.SanMiguel@teamtumbleweed.eu

Abstract

Tracking solutions for satellites orbiting Mars rely almost entirely on stations located on Earth. This kind of tracking service requires scarce and costly communication resources. Furthermore, the service is subject to significant downtime due to occultations produced by Mars itself or third bodies such as the Sun. The use of powerful antennae on both ends is required, making deep space exploration inaccessible. Moreover, the loss of contact produced by an occultation leads to a loss of service. As a result, a prolonged isolation of spacecraft negatively influences its location estimation and scientific data, which can add significant risk to missions.

We propose an alternative tracking service for ongoing and future missions on Mars that enables democratized access to deep space exploration. This service, unlike current solutions, would be delivered from Mars itself, relieving the demand for bandwidth from services such as the Deep Space Network (DSN), and providing service during solar conjunction and other events. We show that the tracking functionalities for Mars missions can, at least partially, be transported to the 'red planet', thus achieving a more efficient usage of the scarce resources. The envisioned infrastructure consists of a network of 50 to 100 beacons to provide tracking services to Mars orbiters. These beacons will be distributed on the Martian surface by a swarm of wind-driven mobile impactors - the Ultimate Tumbleweed Mission (UTM).

In this paper, we show that the proposed system would provide a valid level of accuracy for orbiters around Mars. Simulations are done using open-source orbit estimation software - TU Delft Astrodynamics Toolbox (Tudat). After setting the basic environment, relevant planetary bodies are added as well as a spacecraft to represent the target of our tracking. A trade-off is performed between the number and distribution of beacons, the properties of the instrument (power, frequency, and antenna), and the precision and accuracy obtained, considering the Tumbleweed Science Mission. A comprehensive model for the state-of-the-art tracking methods for Mars orbiters, as well as the proposed network of beacons, are developed and implemented in Tudat. The performance of each beacon can be studied under different circumstances. We also study the most desirable influential noise sources for such a tracking network of beacons. This in-house end-to-end experiment allows us to determine the overall usefulness of a Martian tracking system of the future.

Keywords: Tracking, GNSS, Tumbleweed mission, Mars exploration, swarm

Acronyms

DORIS Doppler Orbitography and Radiopositioning Integrated by Satellite. 4, 14

DSN Deep Space Network. 1, 2, 4, 5, 7

EOL End-Of-Life. 10

MEX Mars Express. 3, 7, 13

S/C Spacecraft. 6

Tudat TU Delft Astrodynamics Toolbox. 5, 10, 11

UTM Ultimate Tumbleweed Mission. 1–3

1 Introduction

Deep space missions, despite usually including autonomous routines in the case of loss of contact, currently rely heavily on support given from the Earth. This is not a big surprise, since the ultimate goal is always for the results to reach the Earth. However, anything beyond science-results related telemetry is, strictly speaking, unnecessary. Moreover, providing such support from Earth is very costly. A single hour of communications using the DSN service costs is in the order of thousands of USD[1]. This poses huge obstacles for space exploration, and inevitably results in very low accessibility to deep space. Performing science beyond Earth is limited to governmental agencies with high funding. To this end, one must add the ever-growing demand and restrictions in DSN budget[2], that only make matters worse.

It is in this context that, in 2017, with the purpose of democratizing access to deep space, the UTM was born[3]. The core development of TeamTumbleweed is a swarm of wind driven rovers/impactors (Figure 1) that transport scientific payload across Mars[4].



Figure 1: The Tumbleweed V3 Prototype rolling across the Negev desert, during the AMADEE20 Mars Analogous Mission [5].

By using off-the-shelf components and by mass producing (~ 90 rovers are planned), the costs can be kept small, as well as most of the risks[5]. Using a lightweight structure that can be folded, transport costs are further reduced[5]. Thanks to the high number of rovers, and their planned retractable sails, an unprecedented spatial coverage of Mars could be attained[5].

This paper proposes using such a distribution of rovers (from now on referred to as *beacons*) during their static phase[5] so as to provide tracking to Mars orbiters, thus providing a cheaper alternative to the Earth based services aforementioned.

Several beacon system configurations are therefore tested and compared, including the possibility to calibrate the system.

2 Background Information

In this section we introduce the different concepts needed to understand the simulations and their results.

2.1 Satellite Orbit Determination

The field of satellite orbit determination describes methodologies to obtain the ephemeris (position and velocity) of a satellite with regard to time. To do so, a dynamic model is normally used, such that given the initial conditions, the ephemeris can be obtained at any t . Such dynamic model must represent the real world as good as possible, with whatever accuracy requirements the mission might have.

For instance, *gravimetry* missions require very accurate dynamic models of the gravitational field, as well as other forces with similar magnitudes, in order to isolate and compute each coefficient. Since we are not focusing in any specific science mission yet, a general purpose dynamic model is enough, as long as it reproduces most of the dynamics involved.

For this, the following gravitational effects were considered:

- **Mars:** Degree and order 4 using *Tudat* default coefficients.
- **Phobos:** Point mass.
- **Deimos:** Point mass.
- **Earth:** Point mass.
- **Jupiter:** Point mass.
- **Sun:** Point mass.

Other effects, such as atmospheric and radiation pressure were not considered, to avoid including a satellite geometry and rotation model for now.

To benchmark this dynamic model, it was compared with the *spice*¹ kernels from the MEX mission[7]. The initial conditions for the dynamic model were obtained from *spice*, and the model was propagated. Figure 2 compares the orbital parameters of both the propagated trajectory and *spice* kernel, after 10 days of simulation.

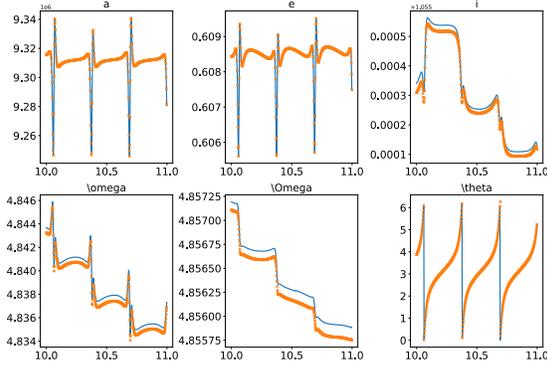


Figure 2: MEX’s simulated (markers) and *spice* (line) trajectory parameters comparison against t

When compared to *spice* kernels, the dynamic model accumulates some errors with time. These errors grow up to $\sim 150km$ after 20 days of simulation.

However, this level of error with regard to the *real* world does not concern our purposes. For now, we mainly focus on benchmarking the system instead of performing actual tracking.

As long as the dynamic model recreates most of the *real* dynamics (which, despite the accumulated error seen in Figure 2, it does), it is sufficient for our intents.

From now on, the satellite *real* dynamics will be provided by the dynamic model. The dynamic model will also be used in the estimation process, completely eliminating the dynamic model performance from the results.

Observation problem

In order to estimate the ephemeris of a satellite, a set of observations is necessary. These observations could be anything related to the satellite’s trajectory.

For instance, in theory, the solar panel temperature can be used to determine a satellite’s position, since the relative position between the satellite, the planet and the Sun is what ultimately produces such changes in temperature.

However, the most common observables are produced using electromagnetic signals, usually using radio frequencies, but also laser technology.

Range (distance between link ends) and range-rate (change of range) are the simplest and most spread ob-

servations types. It is important to note that neither range nor range-rate are directly observable. Thus, these usually come in the form of Doppler measurements and radio or laser ranging.

Doppler observations measure the change in frequency of a signal produced by range-rate. Laser and radio ranging measure the time a signal takes to reach each end, which is directly related to range.

Both measurement types can be implemented using different architectures.

One-way Doppler, while simpler to execute, incurs on errors due to discrepancies in the clock frequencies of both link ends, which are completely removed by using a two-way architecture.

One-way ranging incurs on errors due to discrepancies in the clock times of both link ends. Two-way ranging does not suffer from this, but suffers from errors from inaccurate clock frequencies, which are also present in the one-way architecture.

The system proposed would generate Two-way Doppler measurements, using a Spacecraft-Tumbleweed-Spacecraft architecture. This is vital to remove clock errors that inevitably occur due to clock instability.

However, to simplify the simulations, direct range-rate observables (one-way instantaneous) are created, and the corresponding level of noise is added. After all, both architectures measure the same effect, but with different dependencies and levels of noise. Using direct range-rate measurements with the level of noise expected in the two-way architecture is a first approximation.

The system proposed does not include ranging measurements. Maintaining the clock stability and system calibration required for ranging is specially hard to do in deep space. In order to keep costs low, this has to be sacrificed. If a solution in line with the UTM was found, it must be studied.

Estimation problem

After acquiring the observations, they can be used to estimate the initial state and, thus, the whole trajectory, via the dynamic model.

One of the most robust and widespread methods is the batch least squares algorithm. At each iteration, the effect of the estimated parameters (initial state and others if present) over the observation residuals² is linearized, and the least squares algorithm is applied.

To linearize the model the first time, an initial guess is needed.

¹SPICE is a system for providing scientists and engineers a wide assortment of space mission geometry.[6]

²In this paper, residuals comprise the difference between the observations used to estimate and the ones generated by the estimator at each iteration. Residual, in singular, is the *root mean square* of the residuals.

Current solutions

As mentioned before, deep space missions currently rely on Earth infrastructure to produce tracking-related observations.

DSN is one of the systems in charge of providing support to such missions. It counts with three stations, strategically located in Madrid, Goldstone and Canberra. Each with one 70m antenna and 4 or 5 other telescopes[8].

Another, completely different, tracking system is DORIS. This system provides services to satellites orbiting Earth, and does so using a distribution of beacons. The system proposed in this paper is similar to DORIS, but on Mars.

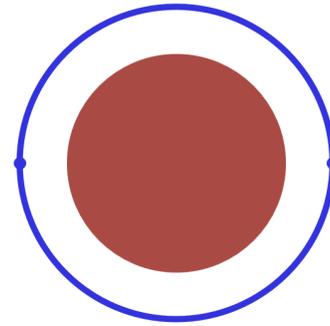


Figure 3: Sensitivity example orbit as seen from Earth (blue dots represent two satellites)

2.2 Accuracy & sensitivity

One of the reasons to migrate the tracking efforts to Mars is the increased sensitivity this results in. When observing martian satellites from Earth, the most significant effect perceived is that of the movement of Mars. It is the variations from that movement that indicate what kind of trajectory the satellite is experiencing.

Providing tracking from such a large distance results in the loss of certain dimensions. Doppler observations made from Earth suffer from rotational symmetry, which means that rotating an orbit around the Earth-Mars direction does not initially affect the measurements. Thus, it is the Earth-Mars geometry that plays the biggest role in the observations, rather than the spacecraft's orbit itself.

Of course, the change of the Earth-Mars direction, and other effects, will affect each of those trajectories differently, given enough time.

On the contrary, if the tracking antennae are located on Mars, every piece of information, without considering its accuracy, is *significant*³.

Sensitivity example

Lets assume two satellites in a circular orbit, with the orbital plane being perpendicular to the Earth-Mars direction, like shown in Figure 3.

Since the position vector (w.r.t Mars) of the satellites is perpendicular to the Earth-Mars direction, their range-rate (w.r.t Earth) should initially be the same as Mars, and their range (w.r.t Earth) slightly higher, but also constant.

All of this breaks down when the Earth-Mars direction starts to change, and the orbit starts to precede. However, for a small period of time, the observations made of both satellites should be the same, despite them being in opposing points in the orbit.

As expected, observations⁴ made using DSN seem to be almost the same for both satellites, as shown in Figure 4. Here, range-rate values for both satellites (green and blue) are superimposed to Mars' Doppler signature (red).

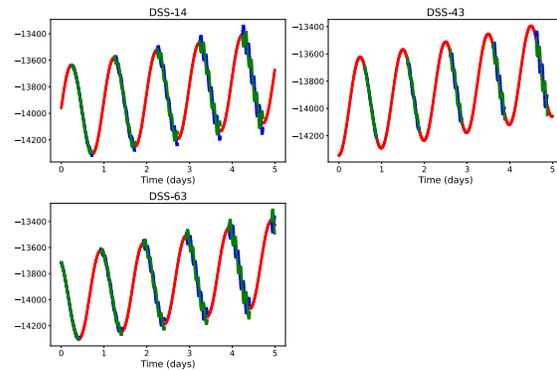


Figure 4: Doppler signature of Mars and two satellites in the test trajectory as seen from DSN

The daily oscillations in all signals are caused by Earth's rotation, and the spans in which each station loses visibility of the satellites is due to Earth obstructing the way.

Removing the effect from Earth rotation and the Mars-Earth relative movement, as well as the visibility restrictions, renders Figure 5. Here it is more clear that both satellites move differently, and the difference between their Doppler signals after 5 days is around 150^m/s.

³In this paper, significance is defined as the sensitivity towards changes in the initial parameters.

⁴These observations are noiseless and perfectly represent the range-rate values.

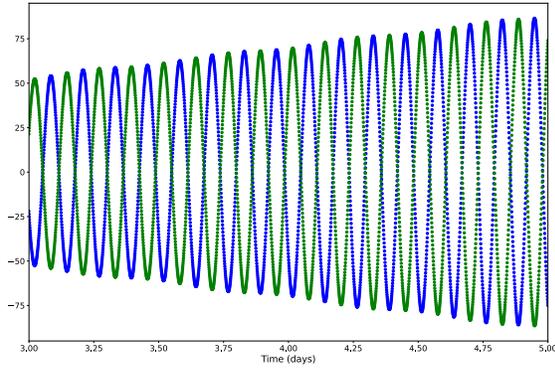


Figure 5: Contribution of the trajectory to the Doppler effect with respect to Earth for both satellites

However, repeating the same analysis using a distributed system on Mars, results in much better observations, as shown in Figure 6. Here, not only are the differences more visually clear, but the numeric value easily reaches 4000m/s

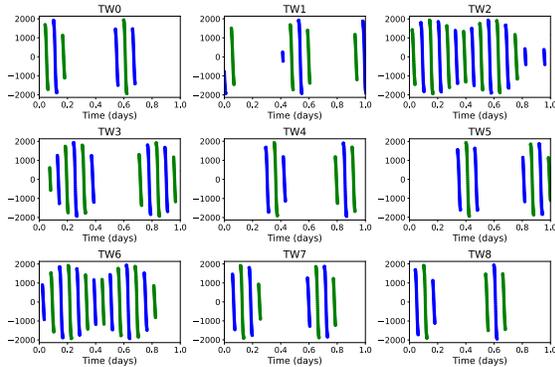


Figure 6: Doppler print from both satellites using TW system (9 beacons)

In the proposed system architecture, the inherent directionality from DSN is lost, resulting in more significant measurements.

This, of course, is just an example, but any trajectory that locally behaves like the one showcased will inevitably produce measurements of low-significance.

2.3 Tudat

The *TU Delft Astrodynamics Toolbox* is a set of tools for simulating different astrodynamics results. It is being actively developed by a community of students and researchers as an open source project. Features are implemented in C++ but, for ease of use, a python wrapper (*Tudatpy*) exists.

⁵All noise was modelled using a Gaussian distribution.

This project is entirely built using *Tudatpy*, as well as other non-astro *Python* dependencies.

To simulate observations, Tudat requires some setup. This comprises, at least, the definition of ground stations and the links between them and the satellite, the nature and timestamps of the observations, as well as viability conditions (elevation angle, occultation, etc.).

During the estimation phase, a dynamical model will be required, since new trajectories using different initial states must be generated.

3 Methodology

To benchmark the proposed system, a set of simulations were conducted using Tudat. The observables used can be divided between range-rate and range.

As explained in section 2.1, range-rate measurements were created using one-way instantaneous Doppler observables with the adequate level of noise⁵.

For range, one-way range measurements with the respective noise level was used. This was again done to simplify the process and for debugging reasons.

In future applications, *averaged* observables should be used instead.

As for viability conditions, all simulations used a minimum elevation angle (ϵ) of 15° , and a 5° avoidance angle with the Sun was imposed. For observations made from the Earth (using DSN), the Earth itself and Mars were used as occultation bodies, while for observations made from Mars, only Mars was used as occultation body.

3.1 DSN model

For every case studied, the 70m telescopes from each station were used (3 antennae in total). Their coordinates were obtained from Tudat default values, and no positioning errors were included.

For range-rate observables a Gaussian noise of 0.2mm/s was used, and for range measurements a Gaussian noise of 1m was introduced.

3.2 Tumbleweed system model

Distribution

Four different distributions were studied: two of them being “ideal” and the other two being more realistic, considering the Tumbleweed mission.

The first “ideal” distribution is a *Fibonacci* distribution, which distributes points on the surface of a sphere almost perfectly uniformly, as seen in Figure 7.

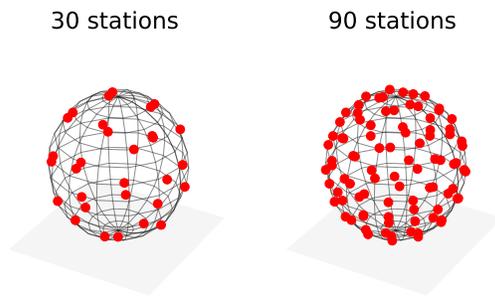


Figure 7: Fibonacci distribution

The other “ideal” distribution, depicted in Figure 8, is just a random distribution, that ensures equal probability for a point to be placed in any semi-sphere.

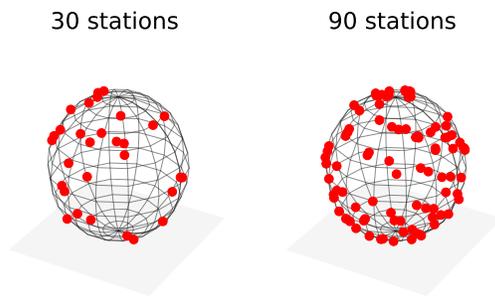


Figure 8: Random distribution

The *polar* distribution (Figure 9) portrays a scenario in which the rovers did not reach too far from the landing site. The distribution depends on a spread parameter (σ), that dictates how spread, in terms of latitude, the beacons are.

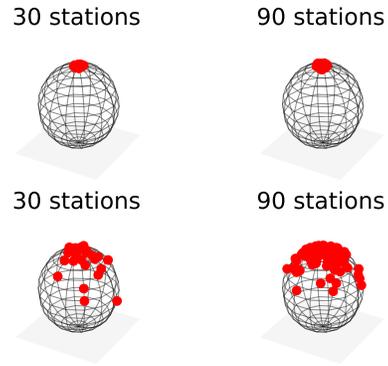


Figure 9: Polar distribution with $\sigma = 5.0$ and $\sigma = 30.0$

Finally, Figure 10 shows a scenario in which the rovers managed to reach the equator. Just like with the *polar* distribution, a spread parameter is defined, which determines the variability in latitude for the beacons.

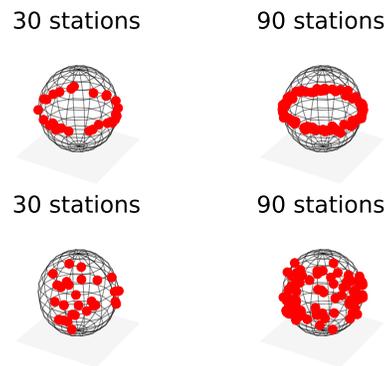


Figure 10: Equatorial distribution with $\sigma = 5.0$ and $\sigma = 30.0$

To account for the uncertainty in the beacons position, a default error of $0.1m$ was introduced between generating the observations and estimating the parameters. This parameter will be referred to as Δx . It represents the standard deviation of a normal distribution centered in each coordinate. Thus, positioning error is added to each coordinate using a $N(0, \Delta x)$ distribution.⁶

Noise

The system proposed would generate two-way Doppler measurements with a S/C-Tumbleweed-S/C. It is assumed that the S/C has a greater power budget and, thus, the noise is mostly generated in the Tumbleweed-S/C link.

As a first approximation, the variance of the range-rate from Doppler measurements is produced by white thermal

⁶This means that the expected distance between a beacon’s real and know location is ~ 1.6 times larger than Δx .

⁷White noise was modeled using $N_0 = kT_{sys}$, and the power to reach the satellite was obtained by applying free-space path loss to the emitted power.

noise. The effect of such noise is shown in Equation 1[9].

$$\sigma_V = \frac{c\sqrt{2B^{N_0}/S}}{4\pi GfT} \quad (1)$$

Where f is the base frequency, G is the transponding ratio, T is the integration time for Doppler measurement, B is the noise-equivalent bandwidth and S/N_0 is the signal to noise spectral density.

The following were used as reference values: $f = 400MHz$ [10], $G = 1.1$, $T = 2s$ and $B = 500kHz$.

Equation 2 defines the signal to noise spectral density ratio in terms of the system and mission parameters⁷.

$$\frac{S}{N_0} = \frac{P}{kT_{sys}a_0M} \quad (2)$$

In our case P was assumed to be $12.9W$ and $T_{sys} = 135K$ [10]. Moreover, for MEX a_0 (free-space path loss factor) is approximately $2.22 \cdot 10^{19}$ and a margin M of 1.1 was used.

This results in a velocity standard deviation of $\sigma_V = 0.028652m/s$. As a measure of safety, a value of $0.05m/s$ was used instead to generate the Gaussian noise. To avoid confusion with the spread parameter (σ), the range-rate noise will be referred to as $\Delta\dot{\rho}$.

4 Results

For all these cases the initial state for *Mars-Express* at epoch 2004 APR 10 12:00:00 UTC obtained from *spice* was used. All the estimations apply to the initial guess an error of $1000m$ in the radial direction.

4.1 DSN

Range-rate

Besides the default system parameters previously mentioned, some simulation variables must be selected: observation time span and observation frequency.

Observation time span determines the period of time in which observations are generated to then be fed to the estimator. In order to select it, different observation time spans were tried, as shown in Figure 11.

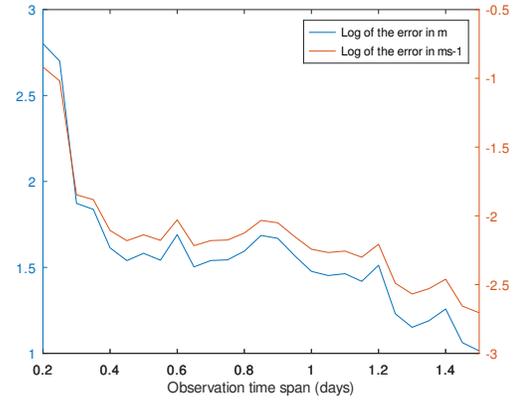


Figure 11: Initial state rms after estimation

As can be seen, the result improves while increasing the observation time span. However, after 1.5 days the solution ceased to converge. Too long observation time spans can make it harder for the estimator to find the solution, so a fixed time span of 1 day was used in all simulations.

After estimating using the default system parameters, an observation time span of 1 day and an observation frequency of $60s$, the observation residual obtained was 0.000200815 (very close to the expected $2 \cdot 10^{-4}$), and the initial state position error was $51.949m$, while the velocity error was $0.00991m/s$.

Studying the observation residuals over time yields Figure 12. It is clear here that the parameter estimation reached the limit imposed by the noise.

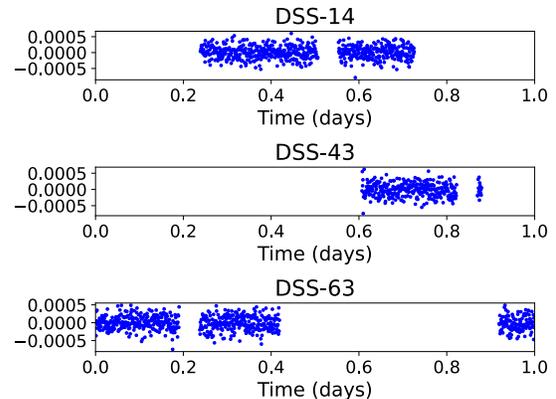


Figure 12: Observation residuals for each antenna

The time spans with no observations correspond either to Earth obstructing the way ($\sim 12h$) or Mars being between the station and MEX ($\sim 1h$).

Recreating Figure 12 without the noise (but the same estimated parameters) yields Figure 13. There it is clear that the solution found is not the *real* solution. The noise

completely hides this behavior, limiting how good the solution found can be.

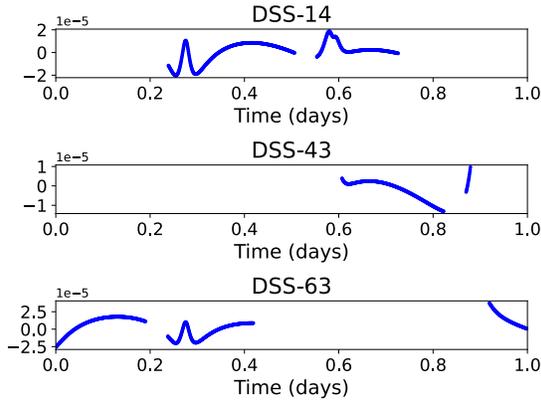


Figure 13: Observation residuals for each antenna without noise

To reduce the errors in the initial state, the observation frequency can be increased to 1s. This results in a similar residual, but an initial state error of just 1.7697m in position and 0.00033812m/s in velocity.

It is important to note that the residuals for both parameter estimates are very similar. Using the observation frequency of 1s, the estimate obtained with the 60s observation frequency has a residual of 0.000211775, while the solution found with the frequency of 1s has a residual of 0.00020019.

Plotting the residuals from the latest estimation gives Figure 14, where again it is clear that the estimation reached the limit imposed by noise.

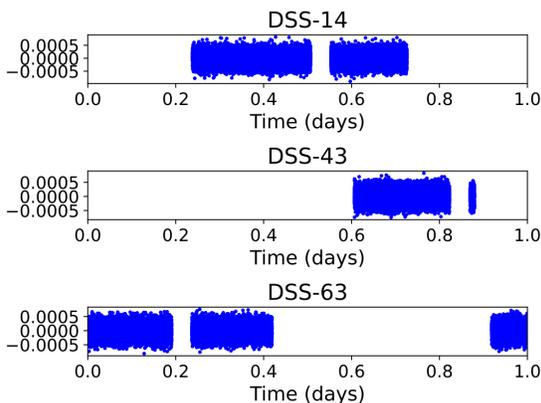


Figure 14: Observation residuals for each antenna (1s case)

Just like before, by removing the noise Figure 15 is obtained. Here, again, it is clear that, due to the noise, the

estimator is missing some different dynamics between the estimated solution and the *real* one. However, thanks to the increased observation frequency, the difference is now one order of magnitude lower.

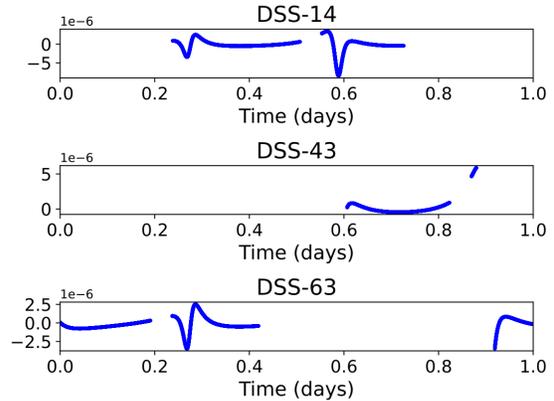


Figure 15: Observation residuals for each antenna without noise (1s case)

By increasing the observation frequency, effectively, the noise frequency is also increased, making it easier for the estimator to differentiate the real dynamics from the noise.

Batched range-rate

A common method to estimate for longer periods of time without failing to converge is to make several chained estimations. Usually, each observation time span starts before the previous one ended, so that both can be compared.

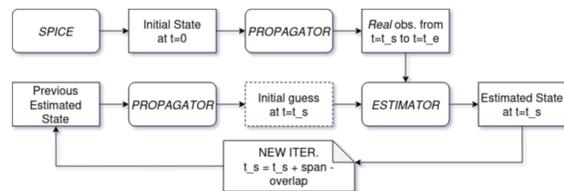


Figure 16: Flow chart for the batched estimations

In this case, observation time spans of 1 day were used, with each one overlapping 0.2 days with the previous and next ones.

The initial guess state for each iteration is obtained by propagating the previous result to the corresponding timestamp, as seen in Figure 16.

Applying this technique to our case, while keeping the 1s observation frequency results in Figure 17, which was generated by subtracting the *real*⁸ ephemeris from each estimated period. It is clear here that, in this case, Doppler

⁸In this paper, *real* is used to refer to propagated dynamics using the initial states from *spice*.

measurements are much more sensitive to variations in the radial position (r), followed by the along-track position (s) and finally the cross-track (w).

In this case, Doppler measurements made from Earth are quite sensitive to changes in the orbit, but not as sensitive to changes in the position inside the orbit, and have very little sensitivity against variations of the orbital plane.

Comparing the trajectory results with each other (in the overlaps), instead of comparing them with the *real* values, yields Figure 18.

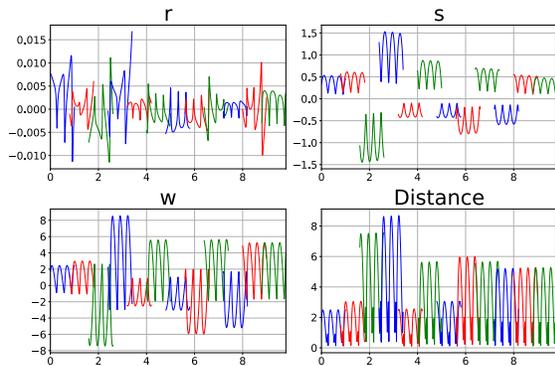


Figure 17: Position difference between batched DSN estimation and *real* values

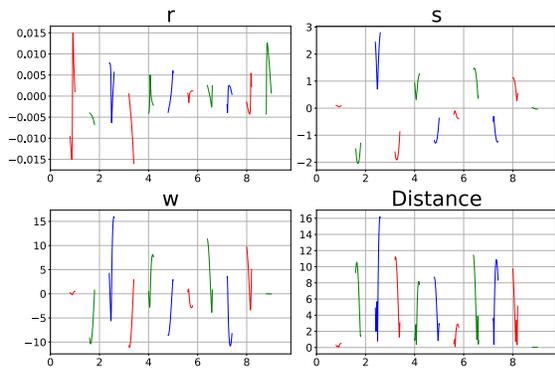


Figure 18: Position difference between each DSN estimation and previous estimation values

Here the error in distance between consecutive estimations is over $10m$, indicating that the *real* error must be of the same order, as can be seen in Figure 17.

Normally, when using real-life observations, one does not know the real states, so Figure 17 can not be created. Instead, Figure 18 is used to determine the tracking performance.

In this case, we can intuitively know what estimations have the highest error. Knowing this, the solution can be improved by making small changes to the corresponding

observation subsets, like starting at a different time, or using a different observation time span.

Range-rate & range

Repeating the batched estimations, but including range measurements results in Figure 19.

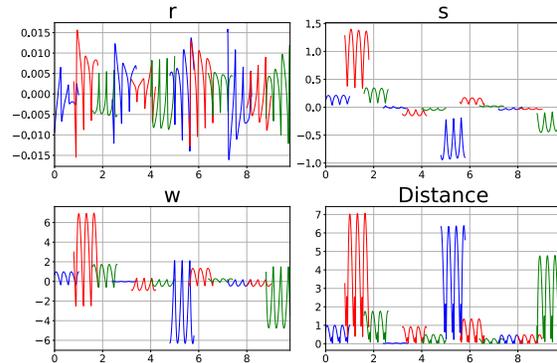


Figure 19: Position difference between batched DSN estimation and *real* values (range-rate & range)

It is clear here that the system performs noticeably better when including range measurements. However the maximum error is also around $10m$.

Just like before, Figure 20 is created by comparing the estimation results with each other. In this case, though, it is even more clear what estimations inside the batch hold the highest errors, since they differ from the rest the most.

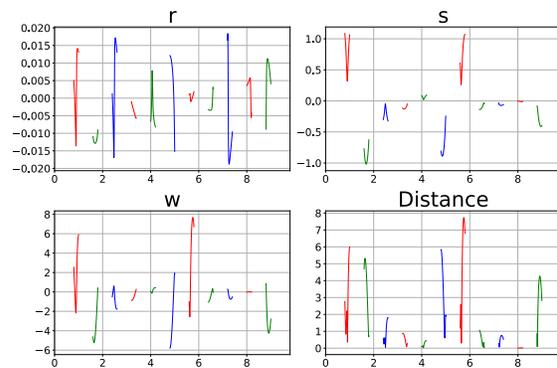


Figure 20: Position difference between each DSN estimation and previous estimation values (range-rate & range)

4.2 Tumbleweed system

Besides the previously defined default parameters, in every simulation, unless stated otherwise, 90 beacons, an observation time span of 1 day and observation frequency of 10s were used.

To avoid random biases, each simulation was performed 48 times. Each time randomizing the beacons' position (if applicable), the beacons' positioning offset (obtaining the change in coordinates using Δx) and the noise (generated using $\Delta \dot{\rho}$). This number of runs was deemed enough, since average values of 48 samples differed less than 10%, or one decimeter per meter.

Range-rate

Using all the default parameters resulted in Figure 21. Each graph in this figure represents one of the distributions previously described in section 3.2, and each marker is a different run of the same simulation.

Since noise is randomly generated, as well as the positioning errors and beacon real positions (except in the *Fibonacci* distribution), each run results in a different residual and initial error.

All of the simulation runs had an observation residual of $\sim 0.05m/s$, which is the observation noise value used. This means that convergence occurred always in every distribution and, thus, the estimator is functioning properly.

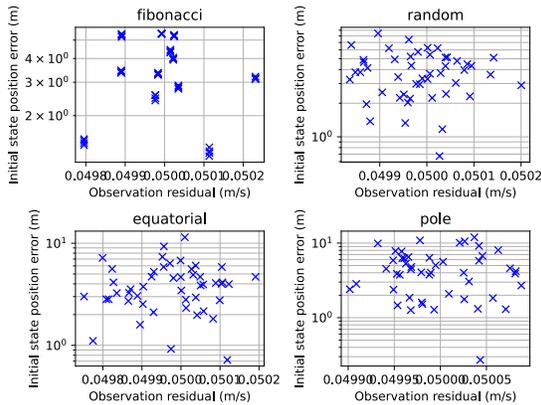


Figure 21: Initial state error vs. observation residuals in default case

In terms of performance, all the distributions seem to provide errors between 1m and 10m, the *Fibonacci* distribution being slightly better than the others.

It is also worth noting that, since in the *Fibonacci* distribution the position of the beacons is not randomized, some simulation runs gave almost same results, which is clearly noticeable in the graph, where markers are bundled in small groups.

This was the result from parallelizing the simulation jobs and Tudat using a fixed random generator seed. To avoid this, a different parallelization technique was used, making the results more variate.

⁹*Fibonacci* and *random* distributions do not have such a parameter.

This particular circumstance lets us compare the effect noise and positioning uncertainty have in the end result. Using the same noise function, even if the beacons' known position was different, resulted in approximately the same solution. This suggests that the effect of noise is greater than that of the beacons' positioning.

In simpler terms, the level of noise introduced produces changes in the solution found much greater than the beacons' position uncertainty introduced.

In order to evaluate the system performance under other circumstances, a sensitivity analysis is performed.

The system parameters are changed (one by one) using the updated values from Table 1.

The alternative value for N represents a situation in which only a fraction of the initial rovers remains active, during the system EOL.

The alternative value for σ represents a situation where, either the rovers could not travel great distances since landing (*polar* distribution), or the rovers finished the rolling phase very close to the equator (*equatorial* distribution).

As for the alternative value for f , it represents a case in which storage, data bandwidth or any other system limitation restricts the rate of observations.

The alternative value for Δx represents a situation with less favourable beacon positioning uncertainty.

Finally, the alternative value for $\Delta \dot{\rho}$ represents a system with better measurement capabilities produced by, most probably more available power.

Table 1: Parameter values used in sensitivity analysis

	N	σ	f	Δx	$\Delta \dot{\rho}$
Def. value	90	30°	10s	0.1m	0.05m/s
New value	30	5°	30s	10m	0.005m/s

As for the expected results, reducing the number of beacons should decrease the system performance, since less observations are generated.

Decreasing the spread should also negatively affect the performance of the *equatorial* and *polar*⁹ distributions since the beacons will be closer together, making each individual observation less significant.

Decreasing the observation frequency (more time between observations) should result in lower performance, as the total number of observations is reduced. Moreover, it can be argued that the value of each observation also depends on the ones surrounding it, since it is the combination of observations what gives a complete picture of the satellite dynamics. Very low observation frequencies (high time between observations) would make impossible to identify these dynamics.

Increasing the beacon positioning uncertainty should result in worse performance, as the estimator will have a harder time fitting the model to the observations made.

Finally, decreasing the range-rate noise should improve the results obtained, as a higher level of detail is visible to the estimator, which then can further improve the solution, as demonstrated in section 4.1.

First, the *Fibonacci* distribution's sensitivity towards the parameters was studied. Figure 22 combines the results from all the cases (except changing the observation noise) using the *Fibonacci* distribution.

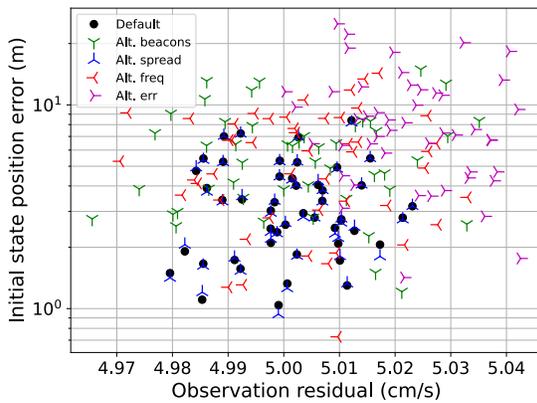


Figure 22: Incomplete parameter sensitivity for the Fibonacci distribution

Here, the alternative spread case provides the most similar results to the default one. This, as stated before, makes sense, since spread does not affect the *fibonacci* distribution, it having fixed beacon locations.

On the contrary, the decrease of beacons, the decrease of observation frequency and the increase in beacon position uncertainty all worsened the system performance, just as expected. However, for all cases initial state errors were kept $\sim 10m$.

It is interesting to note that, with the higher positioning uncertainty, the markers cease to be bundled in groups, since now the beacons are not positioned close enough to render the same result.

Figure 23 includes the results from the alternative range-rate observations noise. Reducing this parameter increased the system performance, as expected.

¹⁰Even though Δx could seem to have a bigger effect, it barely doubles the error, while $\Delta \dot{\rho}$ decimates it. Negative percentual variations are generally larger than their positive counterparts.

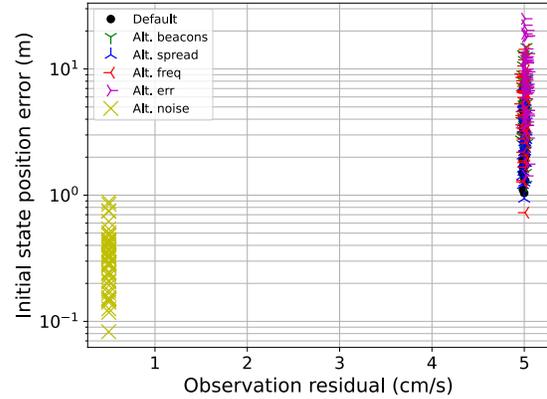


Figure 23: Full parameter sensitivity for the Fibonacci distribution

Table 2: Effect of the parameters on the Fibonacci distribution average performance

	N	σ	f	Δx	$\Delta \dot{\rho}$
Default (m)	3.41				
New (m)	6.10	3.40	5.85	9.02	0.35
New./Def. -1	0.79	0.00	0.72	1.65	-0.90

In fact, the effect from changing the noise was the highest among the parameters. With the reduced noise level, the system produced initial state estimates with lower errors of $\sim 1m$. The observation residual was also decreased, since (when the estimation converges) it tends to match the level of noise.

These results are summarized in Table 2. Here, σ has the least proportional effect, and $\Delta \dot{\rho}$ the highest¹⁰. In fact, the only difference between the simulations performed with the default values and with the updated σ value lies in the exact beacon positioning errors used, which are randomized each time.

As explained before, due to Tudat's architecture, the same noise is used for each observation. Each simulation case (combination of distribution and parameters) was run 48 consecutive times. Each of those 48 times the noise introduced by Tudat is different, but the same run among different simulations uses exactly the same noise.

This means that, for instance, each of the default runs has a one to one relationship with each of the updated σ runs in terms of using the same noise function. This results in the markers from each of those cases being very close from a marker of the other case, as seen in Figure 22.

Following with the *random* distribution, and just like before, Figure 24 was produced.

Again, and as expected, the default case and the alternative spread case perform the closest. However, in this case the difference is larger. This is due to the inherent position randomization caused by the *random* distribution, that makes simulation runs more different to each other.

Just like with the *Fibonacci* distribution, the decrease in beacons, decrease in observation frequency and increase in beacon position uncertainty negatively affected the performance of the system. However, the initial state errors were kept $\sim 10m$.

Unlike with the *Fibonacci* distribution, in this case there is no coupling between the default case and the alternative spread case. This is because the beacons' position is randomized for every run, making them always different. This is just the opposite of the *Fibonacci* distribution, where the beacons' *real* position is always the same, even if their *known* positions change, due to beacons' positioning uncertainty (Δx).

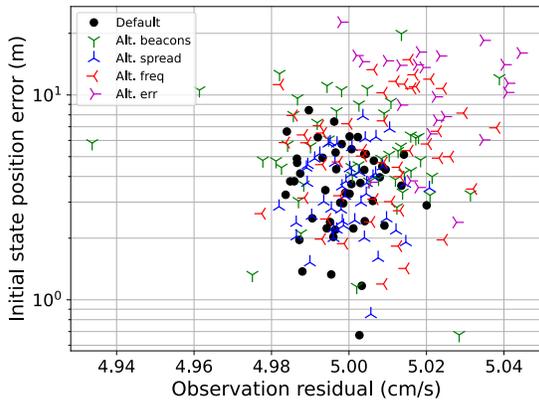


Figure 24: Incomplete parameter sensitivity for the random distribution

As shown in Figure 25, the noise reduction again had the greatest effect on the system. Initial state errors of $\sim 1m$ were obtained.

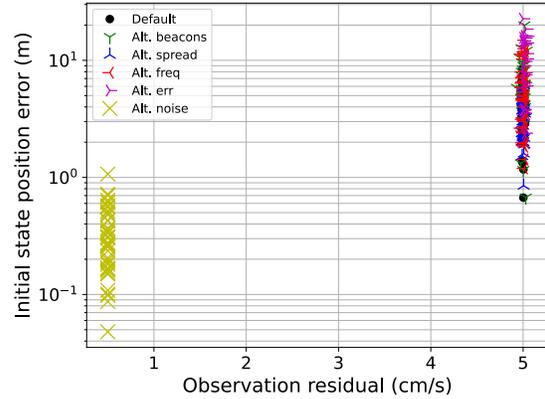


Figure 25: Full parameter sensitivity for the random distribution

Table 3: Effect of the parameters on the random distribution average performance

	N	σ	f	Δx	$\Delta \rho$
Default (m)			3.88		
New (m)	6.06	3.65	6.17	11.30	0.34
New./Def. -1	0.56	-0.06	0.59	1.91	-0.91

Using the same procedure, Figure 26 was generated for the *equatorial* distribution.

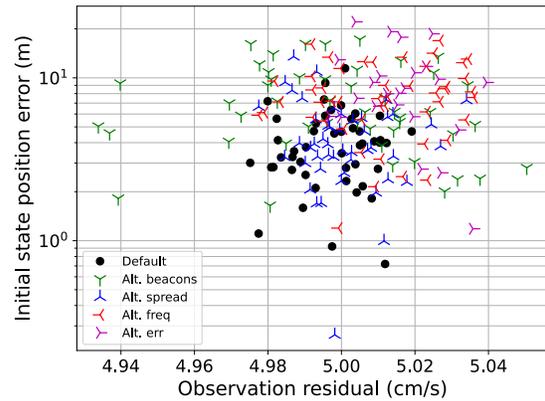


Figure 26: Incomplete parameter sensitivity for the equatorial distribution

Here, while it could be argued that the alternative spread value performs slightly worse than the default one (as seen in Table 4), the effect is similar to the one encountered in the *random* distribution, where we know for a fact that the effect is null, being caused by the randomness of the simulation runs.

The rest of parameters, however, do perform as expected, and in a similar fashion to the previous distribu-

tions, as seen in Table 4.

Just like in the previous cases, Figure 27 includes the effect of reducing the observations' noise, which, again, sets the estimated initial state position error at $\sim 1m$.

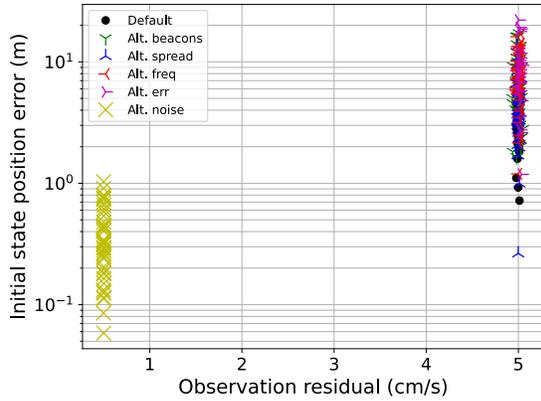


Figure 27: Full parameter sensitivity for the equatorial distribution

Table 4: Effect of the parameters on the equatorial distribution average performance

	N	σ	f	Δx	$\Delta \dot{\rho}$
Default (m)			4.09		
New (m)	7.43	4.54	8.04	9.43	0.40
New./Def. -1	0.81	0.11	0.96	1.30	-0.90

Finally, repeating the process for the *polar* distribution yields Figure 28.

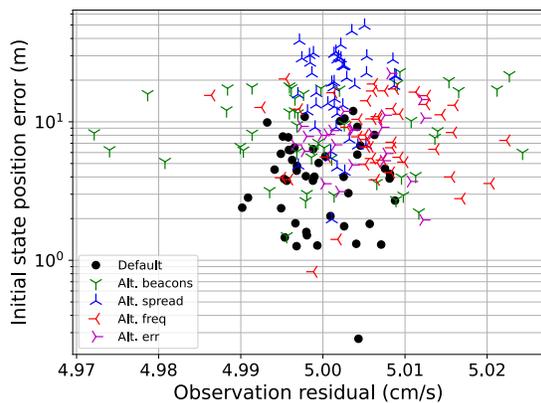


Figure 28: Incomplete parameter sensitivity for the polar distribution

The main difference between this and the other distributions is that σ has a much larger effect on the results. Table 5 shows that, changing $\sigma = 30^\circ$ to $\sigma = 5^\circ$ resulted

in average initial state position errors of almost $20m$, more than 3 times larger than with the default case.

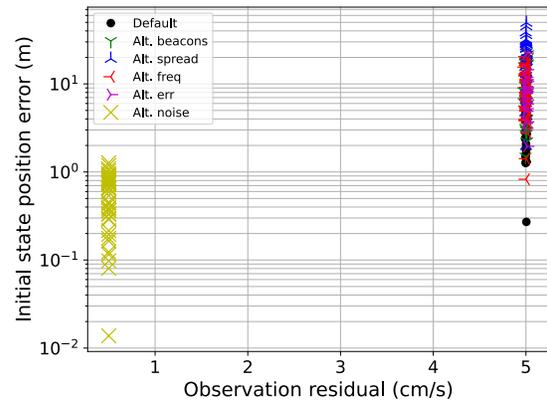


Figure 29: Full parameter sensitivity for the polar distribution

Table 5: Effect of the parameters on the polar distribution average performance

	N	σ	f	Δx	$\Delta \dot{\rho}$
Default (m)			4.72		
New (m)	9.97	19.97	9.23	8.14	0.58
New./Def. -1	1.11	3.23	0.95	0.72	-0.88

Note on the performance of the polar distribution

MEX's orbit has very high inclination, meaning that each pole has quite good visibility of the orbiter once per orbit.

The number of observations produced was easily twice as the other distributions. During estimation it was noted that this particular distribution presented the highest computational times.

This situation benefits the performance of the *polar* distribution, since it has great visibility of the satellite. If presented with a low inclination orbit instead, the performance of the *polar* distribution would suffer. In extreme cases (very low inclination and very low beacon spread) such distribution could have extremely low visibility of the satellite, if any.

It is also important to know that, despite the higher observation count, the performance of the system using polar distribution was slightly worse than with the others. This suggests that such distribution produces less *significant* measurements.

After all, with the *polar* distribution, beacons tend to concentrate in a single point, making their measurements more similar to each other. This is the same effect the spread factor has: the closer the beacons are to each other, the worse the estimation results are.

On the contrary, the *equatorial* distribution actually presented the lowest observation count, but performed (slightly) better.

Range-rate with DORIS settings

By replacing the default noise with 0.4mm/s [11] and removing the positioning error, Figure 30 was created.

All distributions perform similarly, with errors in the order of centimeters, just like the DORIS system on Earth[11].

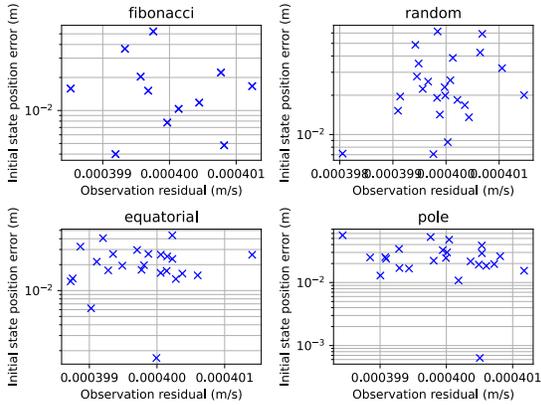


Figure 30: Initial state error vs. observation residuals in doris case

Range-rate & repositioning

Just like with DORIS, the system proposed can use the measurements to calibrate the beacons' position, besides providing tracking.

This case study comprises a single stranded beacon that happens to be $\sim 160\text{m}$ (Gaussian error with $\sigma = 100\text{m}$ was introduced in each coordinate) away from its supposedly known location. The rest of the system is assumed to be in nominal conditions.

Two sets of simulations were performed using the parameters from Table 6, each set including simulations using each proposed distribution. A observation time span of 5 days was used for all simulations.

Figure 31 shows the results from those simulations. As before, all the distributions performed similarly, obtaining an accuracy of 10m for *Case 1* and of 1m for *Case 2*.

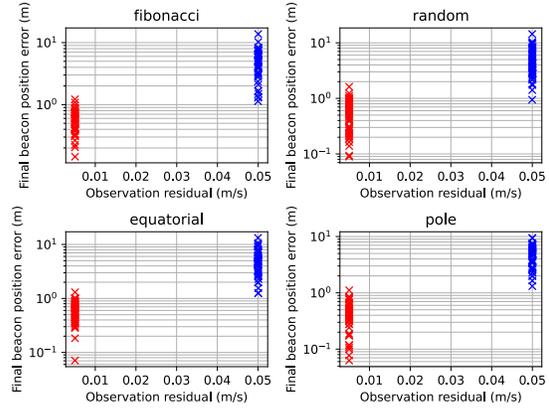


Figure 31: Positioning estimation performance for Case 1 (blue) and Case 2 (red)

Table 6: Parameters for the re-positioning simulations

	N	σ	f	Δx	$\Delta \dot{\rho}$
Case 1	90	30°	10s	1m	0.05m/s
Case 2	90	30°	10s	1m	0.005m/s

5 Discussion

The simulations performed behave as expected, and the effect of the parameters is consistent with the theory. Similar results to those claimed by the real DORIS network were obtained using real parameters, indicating the validity of the methodology.

5.1 Tracking

The performance of the Tumbleweed system was consistent, with the *polar* distribution being the least accurate. Initial state estimations had an error of $\sim 10\text{m}$ for most of the cases (including the default case).

Out of all the parameters studied, range-rate noise had the most overall effect, with the exception of spread in the *polar* distribution.

Keeping the beacon positioning uncertainty at a similar level as the initial state estimation error had little effect on the results. However, simulations performed with a higher level of Δx resulted in estimated initial state errors of $\sim \Delta x$. This indicates that, while the estimated initial state error is $\gtrsim \Delta x$, the system is bottle-necked by a different parameter.

Thus, the system's performance using the default parameters was mainly limited by the noise level.

5.2 Positioning

As for the repositioning of beacons, the system behaved quite well, being able to locate a beacon $\sim 100\text{m}$ away

from its supposedly known location with $1m - 10m$ accuracy, in just 5 days.

Using an improved system (Case 2), resulted in accuracies of $0.1m - 1m$.

6 Conclusion

After determining the relevant parameters and performing the simulations, it can be concluded that that a system like the Tumbleweed network of land based navigation beacons could provide tracking capabilities with an accuracy of $\sim 10m$.

This accuracy could be increased by improving the communications architecture (reducing range-rate noise).

The distribution of the beacons seemed to play a secondary role, only being specially relevant when the beacons concentrate in a single point (*polar* distribution). Were the Tumbleweed mission to proceed as planned, this would not be an issue.

The accuracy of the positioning of the beacons also seemed to play a secondary role, and it did not affect the

tracking performance as much as other parameters.

However, such change was noticeable enough for the estimator to improve the positioning of a single stranded beacon from a positioning error of $\sim 100m$ to just over $1m$.

This means that, regardless of the ultimate feasibility of the tracking system, the network could always be used in reverse, to provide accurate positioning of the beacons.

Future research (performed inside the Tumbleweed mission) should include:

- Further system specification & more accurate modelling (communications system definition and beacon distribution[12] can be improved, better models for the noise can be included).
- Further objective definition (satellite tracking, beacon positioning or both, rover positioning during non-static phase, etc.)
- Further development on the science mission case.
- Researching estimation strategies best suited for this mission.

References

- [1] T. Pham and A. Bedrossian, "Deep space network services catalog," *DSN No. 820*, vol. 100, 2022.
- [2] J. Foust, "Nasa deep space network reaches "critical point" as demand grows," *SPACE-NEWS*, 2023. [Online]. Available: <https://spacenews.com/nasa-deep-space-network-reaches-critical-point-as-demand-grows/>
- [3] "Our vision," <https://www.teamtumbleweed.eu/our-vision/>, 8 2023.
- [4] "Our rover," <https://www.teamtumbleweed.eu/our-rover/>, 8 2023.
- [5] J. Rothenbuchner, L. Cohen, F. Abel, D. Buryak, K. Cuervo, J. Kingsnorth, O. Mikulskytė, A. Phillips, M. Renoldner, and M. Sandrieser, "The tumbleweed mission: Enabling novel mars data sets through low-cost rover swarms," in *Proceedings of the 73rd International Astronautical Congress*, 2022.
- [6] C. Acton, "An overview of spice," *Jet Propulsion Laboratory: Oak Grove, KY, USA*, 1998.
- [7] "Mars-express spice kernels," <https://naif.jpl.nasa.gov/pub/naif/MEX/kernels/spk/>, 2020.
- [8] D. S. Network, "Dsn telecommunications link design handbook," *Pasadena, CA: JPL*, 2010.
- [9] P. W. Kinman, "Doppler tracking of planetary spacecraft," *IEEE Transactions on microwave theory and techniques*, vol. 40, no. 6, pp. 1199–1204, 1992.
- [10] F. Abel, C. Ferent, P. Sundaramoorthy, and R. Rajan, "Communications architecture for martian surface exploration with a swarm of wind-driven rovers," in *Proceedings of the 73rd International Astronautical Congress*, 2022.
- [11] ESA, "Doris overview," *Earth Online*. [Online]. Available: <https://earth.esa.int/eogateway/instruments/doris/description>
- [12] M. Kapadia, L. Cohen, M. Renolder, D.-A. Vicovan, F. Mahmood, O. Mikulskytė, and J. Rothenbucher, "Application of a monte carlo-based performance modeling tool in model-based systems engineering of a tumbleweed mobile impactor," in *Proceedings of the 74th International Astronautical Congress*, 2023.