Internship Report

Satellite Communications and Orbital Control

14 November 2020

Author: Yasin Sönmez

Institution: TUBITAK UZAY

Table of Contents

1 Introduction

1.1 TUBITAK UZAY (Space Technologies Research Institute)

TÜBİTAK Space Technologies Research Institute (Turkish: TÜBİTAK Uzay Teknolojileri Araştırma Enstitüsü) or TÜBİTAK UZAY for short, is a Turkish institution carrying out research and development projects on space technology, electronics, information technology and related fields. It was established in 1985, under the name "Ankara Electronics Research and Development Institute" within the campus of Middle East Technical University (ODTÜ) in cooperation with the Scientific and Technological Research Council of Turkey (Turkish: Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, TÜBİTAK) and the university in Ankara. In 1995, the organization was renamed. Since 1998, the institute houses a new building in the campus.

TÜBİTAK UZAY specializes in space technologies, electronics, information technologies and related fields, keeping abreast of latest technological developments. The institute leads and takes part in R&D projects, aiming at having a pioneering role in the national research community, and assisting the industry in solving technical problems encountered during system design, selection and use, product development and manufacturing in abovementioned specialization areas.

TÜBİTAK UZAY gives special emphasis on developing capability on small satellite design, manufacturing and test, leading Turkish Space Program together with Turkish Aerospace Industries and Turkish Satellite Assembly, Integration and Test Center to initiate international collaboration in space technologies. [1]

1.1.1 Research Areas

In TUBITAK UZAY research in the following areas are being made:

Space Technologies: Satellite systems, satellite sub-systems, satellite ground station sub-systems, satellite test and integration systems.

Electronics: Communication systems, electronics system design, electro-optic mission payload, high-speed digital design, IC design.

Software: Computer vision, speech processing, pattern recognition, remote sensing, multimedia technologies, data mining, machine learning, natural language processing, artificial intelligence.

Power Electronics: Power quality, compensation systems, electrical motor drives, switched-mode power supplies, renewable energy resources.

Power Distribution Systems: Analysis of electric production and transmission systems, strategic research and development in distribution automation, Supervisory Control and Data Acquisition (SCADA) Systems, criteria setting for planning, design and operation of distribution systems.

1.2 Satellites of TUBITAK UZAY

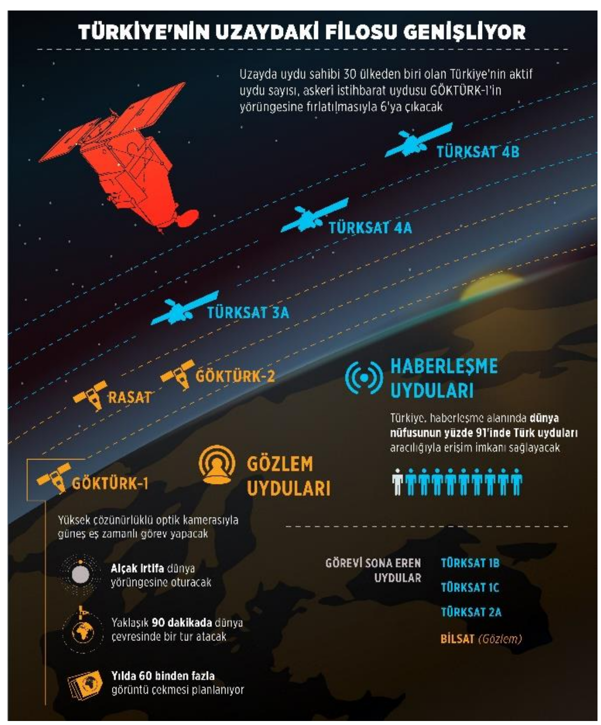
1.2.1 RASAT

Figure 1 Istanbul from the eyes of RASAT [11]

RASAT is an earth observation satellite designed and developed by TÜBİTAK Space Technologies Research Institute (TÜBİTAK UZAY) and produced in Turkey to provide high resolution imagery. It is so the first one of its art completely realized in Turkey, and the second indigenously developed remote sensing satellite after BILSAT-1

Projected for a mission duration of three years, RASAT is on a sun-synchronous geocentric orbit. Its instruments, supplied by the South Korean space technology company Satrec Initiative, allow a spatial resolution of 7.5 m (25 ft) at panchromatic band and 15 m (49 ft) at multispectral band. RASAT carries out various civil applications on mapping and planning, disaster management, ecosystem monitoring, environmental control, landcover survey and coastal zone management. Additionally, RASAT is used to test a custom designed on-board computer "BiLGE" capable of using SpaceWire network, a solid-state processor "GEZGİN-2" (an abbreviation for "Gerçek Zamanda Görüntü İşleyen") for real-time image compression using algorithm of JPEG 2000 and a telecommunication system "Treks" of X band transmitter module with 100 MB/s data transfer rate and 7 Watt power. [2]

1.2.2 GOKTURK 2



Göktürk-2 is an earth observation satellite designed and developed by the Scientific and Technological Research Council of Turkey (TÜBİTAK) and built by TÜBİTAK Space Technologies Research Institute (TÜBİTAK UZAY) and Turkish Aerospace Industries (TUSAŞ) for the Turkish Ministry of National Defence

Produced with 80 percent indigenously developed technology and 100% domestically developed software, the satellite offers high-resolution imagery of 2.5 m (8.2 ft) resolution at panchromatic, 10 m (33 ft) at multispectral (VNIR) and 20 m (66 ft) at SWIR band. It is Turkey's second national satellite following RASAT, which was launched from Russia on August 17 the same year. For the telecommunication, it has three S band receivers and transmitters. Göktürk-2 later put its solar panels into service, and began the week after the launch to send data and its first images, which were from the US, Brazil, India as well as Turkey's western city of Izmir. [3]

1.2.3 Türksat 6A

Türksat 6A is a Turkish communications satellite under construction, which will be developed and produced indigenously to be operated by Türksat. This satellite will be Turkey's first fully domestically-produced communications satellite, and is expected to be completed by 2020 and launched in 2022.

It will be in a geosynchronous orbit positioned at 42° East with an expected on-orbit life time of at least 15 years. It will consist of 16 Ku band transponders, additionally 4 in reserve as well as two active and one in reserve X band transponders. The Ku band transponders will have a bandwidth of 7.3-18.1 GHz for uplink and 11.7-12.75 GHz downlink bandwidth. Each of the Ku band transponders will have a minimum power of 140 W and the X band transponders at least 150 W each.

While the X band transponders will cover the territory of Turkey only, the Ku band transponders of Türksat 6A will have three coverage zones:

- entire Turkey,
- "West Zone" covering the British Isles in the west, Scandinavian countries in the north, North Africa in the south, Caspian Sea in the east,
- "East Zone" covering Anatolia in the west, Russian Federation in the north, Saudi Arabia and Pakistan in the south, China national boundary in the east.

It is estimated that the project will cost about US$250 million. [4]

1.2.4 IMECE

IMECE Program consists of two main projects:

1) IMECE Satellite Subsystems Development Project and

2) IMECE Satellite Platform Development Project.

Based on the heritage and expertise acquired in BİLSAT, RASAT and GÖKTÜRK-2 Projects, IMECE is set to satisfy future sub-meter resolution needs of Turkey as well as to build related infrastructure and capability domestically.

IMECE Satellite Subsystems Development Project was started in 2013 with funding from The Ministry of Development in coordination with The Minister of Defense. These projects composed of several sub-systems such as Electro-Optical Satellite Camera, Communication System, Star Tracker, Sun Sensor, Hall Effect Thruster System, Reaction Wheel, Payload Data Storage, Compression and Formatting Unit, and Next Generation Onboard Computer. IMECE Satellite Platform Development Project was started in 2017 with funding from TÜBİTAK. [5]

2 Communication Systems

2.1 Theory

2.1.1 Fundamentals of Signals

Communications is the transfer of information across a medium using signals of some sort. In general, signal is the time series of a amplitude. On the transmitting side a signal is encoded and on the receiver side this signal is decoded. These steps are performed differently for analog and digital signals.

2.1.2 Signal Types

The signals that are found in nature are in general analog for example sound, light, heat, electronic signals going through the air. Most digital signals are acquired by sampling these continuous signals. A discrete signal is a signal which has values for specific time instants. All wireless signals are analog. A signal may start out analog, converted to digital for modulation, and converted back to analog for radio frequency (rf) transmission and then converted back to digital. Both forms are needed to complete a link. A communications chain is a combination of both of these types of signals. If you work with just the handsets (units) then you will most likely work only in the digital domain. If you work on link design or in very high-powered equipment such as satellites, then analog issues become important. [6]

2.1.3 Carrier Signals

Carriers help to carry information signal to long distances. We need them because in order to receive the signal at the receiver we need high power signal, this is done by transmission antennas that send the signal to a concentrated direction. However, the antenna gain is a function of the square of the frequency, the higher the gain higher the frequency. That's why higher frequency carriers are used. This process of mapping the information signal to higher frequencies is called modulation. However, there is one requirement a carrier must meet: its frequency must be at least two times the highest frequency in the information signal. This property is a direct consequence of the sampling theorem and it's called "Nyquist Rate". [6]

2.1.4 Error Detection and Correction

2.1.4.1 Binary Symmetric Channel

A binary symmetric channel (or BSC) is a common communications channel model used in coding theory and information theory. In this model, a transmitter wishes to send a bit (a zero or a one), and the receiver receives a bit. It is assumed that the bit is usually transmitted correctly, but that it will be "flipped" with a small probability (the "crossover probability"). This channel is used frequently in information theory because it is one of the simplest channels to analyze. [7]

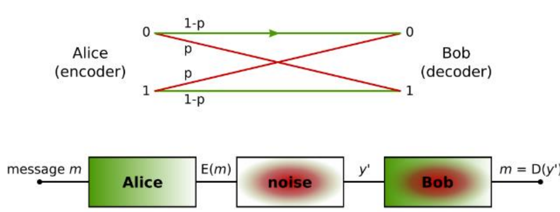


Figure 2 Binary Symmetric Channel model

It is assumed that the crossover probability "p" is smaller than 1/2 because if p>1/2 then it could be flipped at the output to get a probability of flipping lower than 1/2.

2.1.4.2 Error detection and correction

In the transmission of data through a noisy channel the useful information may change. Error detection techniques allow us to detect those errors introduced by channel and error correcting techniques help us to recover back the original data. The idea behind error detection and correction is to add some "redundancy" to the message which will be used by the receiver to check the consistency.

2.1.4.3 Forward Error Correction (FEC)

The sender encodes the data before transmission and adds some redundant bits to data, after the receiver receives the data it applies the corresponding decoding algorithm to correct the wrong bits.

2.1.4.4 Burst Errors

Burst errors are errors that are repeated for many consecutive bits rather than random bits. Examples of burst errors are in general storage devices. For example, scratch on a disk can cause a local error area. If a bit has an error it's very likely to have another error in the next bit.

We can use "block interleaver" to solve this burst error problem. Its aim is to randomize the data by reading in a different order so that we don't have burst errors and the receiver reads back at that format too for convenience.

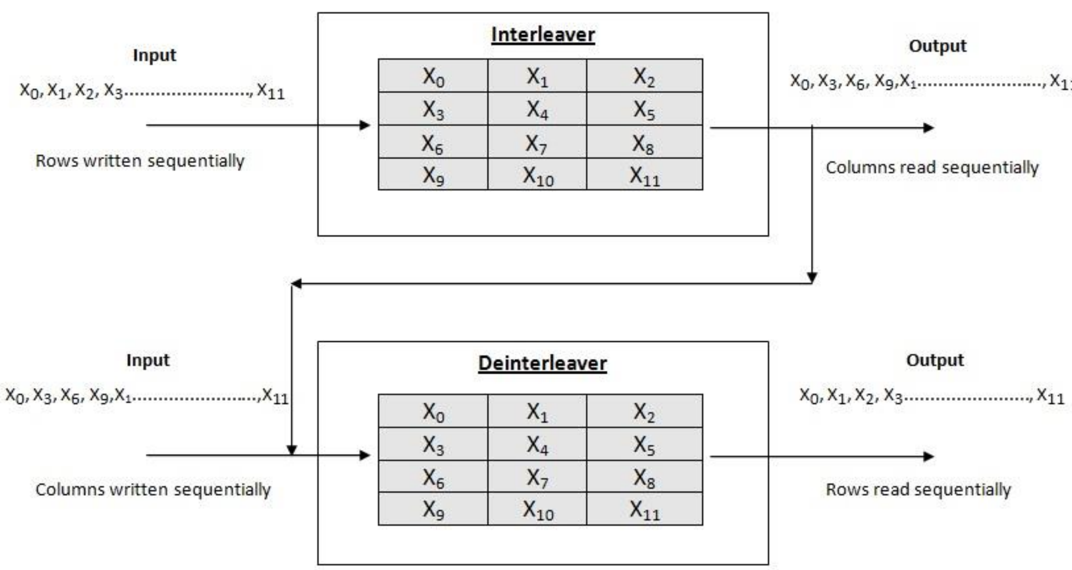
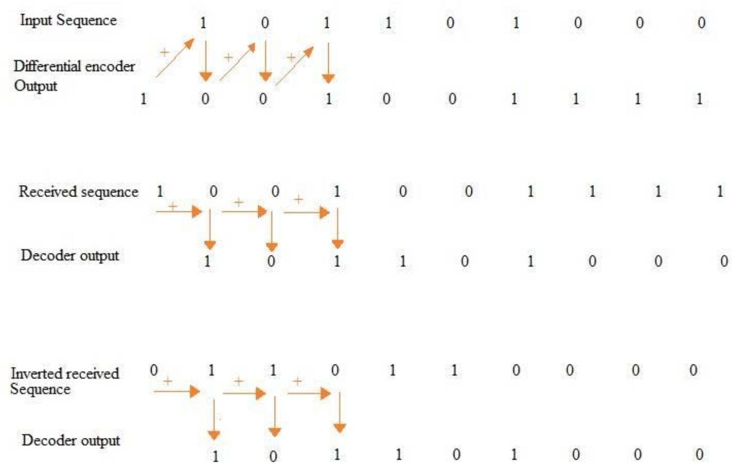


Figure 3 Architecture of block interleaver and deinterleaver [12]

2.1.4.5 All Bits Flipping and Line Coding

Sometimes when the code is going through the media all bits may flip at the same time. So, we need to create a code that can recover from all bit flipping if it happens. We can use line coding for this.

Line code is a pattern of voltage, current, etc. to represent the transmitted information. It may use the past data to encode the data in a format that only keeps the information of if the bit changed or not. This is called “differential coding”. Using this will prevent from all bits flipping error.



Working of Differential encoder and Differential Decoder

2.1.4.6 Repetition Code

A repetition code is a “coding scheme” that sends the information repeatedly for a predetermined time. For example, the code “101” can be sent as “101 101 101” and if somehow some bits are misconducted, majority vote can be counted to select either 1 or 0 for a single bit. However, the repetition code is inefficient. The good thing with repetition code is that it is very simple to implement.

2.1.4.7 Hamming Code

Hamming codes are a family of linear error-correcting codes and they can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors. Hamming codes are perfect codes, that is, they achieve the highest possible rate for codes with their block length and minimum distance of three.

For each $r \geq 2$ there is a code with block length $n = 2^r - 1$ and message length $k = 2^r - r - 1$ hence the rate of hamming code is $R = \frac{k}{n} = 1 - \frac{r}{2^r - 1}$ which is the highest possible for codes with minimum distance three(i.e., the minimal number of bit changes needed to go from any code word to any other code word is three) and block length $2^r - 1$.

An algorithm can be deduced from the following description:

1. Number the bits starting from 1: bit 1, 2, 3, 4, 5, 6, 7, etc.
2. Write the bit numbers in binary: 1, 10, 11, 100, 101, 110, 111, etc.
3. All bit positions that are powers of two (have a single 1 bit in the binary form of their position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)

4. All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.
5. Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position.
    1. Parity bit 1 covers all bit positions which have the **least** significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.
    2. Parity bit 2 covers all bit positions which have the **second** least significant bit set: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc.
    3. Parity bit 4 covers all bit positions which have the **third** least significant bit set: bits 4–7, 12–15, 20–23, etc.
    4. Parity bit 8 covers all bit positions which have the **fourth** least significant bit set: bits 8–15, 24–31, 40–47, etc.
    5. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

If a byte of data to be encoded is 10011010, then the data word (using _ to represent the parity bits) would be ___1_001_1010, and the code word is 011100101010.

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coverage	p1	✗		✗		✗		✗		✗		✗		✗		✗		✗		✗		
	p2		✗	✗			✗	✗			✗	✗			✗	✗			✗	✗		...
	p4				✗	✗	✗	✗					✗	✗	✗	✗					✗	
	p8								✗	✗	✗	✗	✗	✗	✗	✗						
	p16																✗	✗	✗	✗	✗	

Figure 4 Parity bits and their coverage [8]

Parity bits	Total bits	Data bits	Name	Rate
2	3	1	Hamming(3,1) (Triple repetition code)	1/3 ≈ 0.333
3	7	4	Hamming(7,4)	4/7 ≈ 0.571
4	15	11	Hamming(15,11)	11/15 ≈ 0.733
5	31	26	Hamming(31,26)	26/31 ≈ 0.839
6	63	57	Hamming(63,57)	57/63 ≈ 0.905
7	127	120	Hamming(127,120)	120/127 ≈ 0.945
8	255	247	Hamming(255,247)	247/255 ≈ 0.969
			...	
m	$n = 2^m - 1$	$k = 2^m - m - 1$	Hamming$(2^m - 1, 2^m - m - 1)$	$(2^m - m - 1)/(2^m - 1)$

Figure 5 Number of parity bits and the corresponding rate [8]

When the code is received, the receiver can check the correctness of the parity bits and the sum of the positions of the erroneous parity bits identifies the erroneous bit.

2.2 Results

2.2.1 BSC

In this section, we investigate the random bit flipping without any additional specification of errors such as burst error or all bit flipping. All bits have error probability P_e. The resulting error probability at the output of the decoder is P_s.

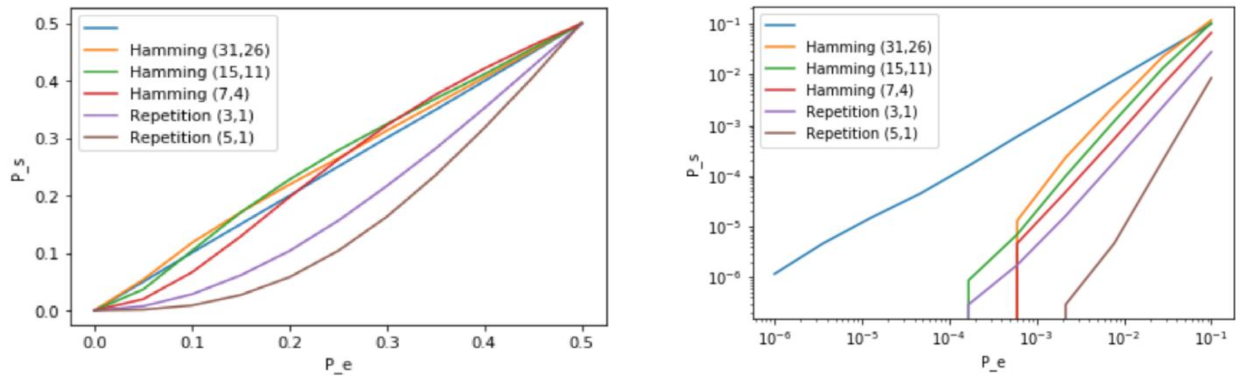


Figure 6 Error rates of different methods. Linear scale at left and log scale at right

2.2.2 BSC + Burst Error

In addition to the before model, we add another layer of burst error so that the error is repeated for a number of times. We used "block interleaver" to recover the original data.

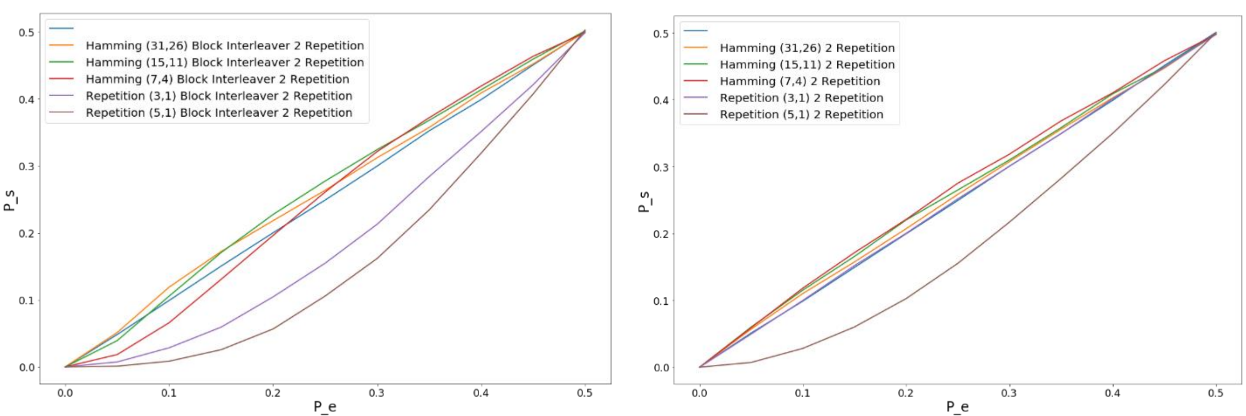


Figure 7 Error rates for burst error of 2 repetition. Interleaver is used in the left and not used in the right one.

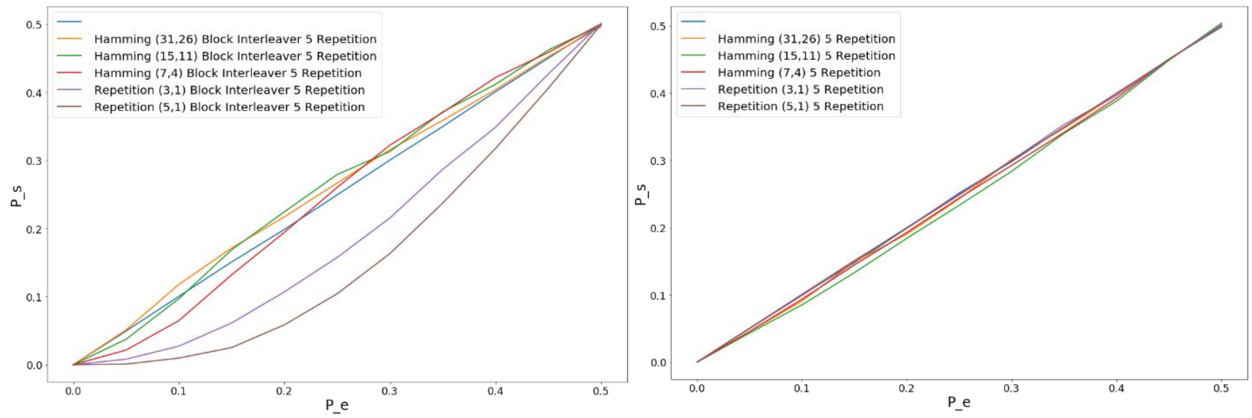


Figure 8 Error rates for burst error of 5 repetition. Interleaver is used in the left and not used in the right one.

2.2.3 BSC + Burst Error + Bit Inversion

Now we investigate the bit inversion case. All the bits may flip at the same time. We use the NRZ-M line coding to get rid of the effect of bit inversion using differential coding.

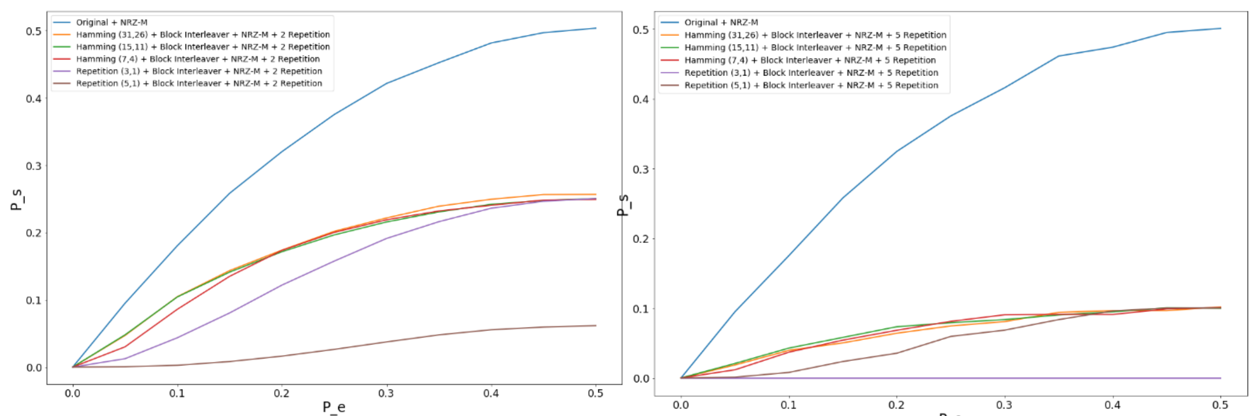


Figure 9 Error rates for bit inversion case using NRZ-M line coding. 2 repetition on the left and 5 repetition on the right

3 Control Systems

3.1 Theory

3.1.1 Orbits in 3D

To define an orbit in a plane we need to have two parameters: “Eccentricity” and “angular momentum”. To define a point in the orbit we need one additional parameter say “true anomaly”. If these parameters are given, we know where exactly a point is at any time in the future.

If we increase the number of dimensions by one and move to 3D, we need additional 3 parameters to describe the orientation of orbit. The name of these parameters is “Euler Angles”.

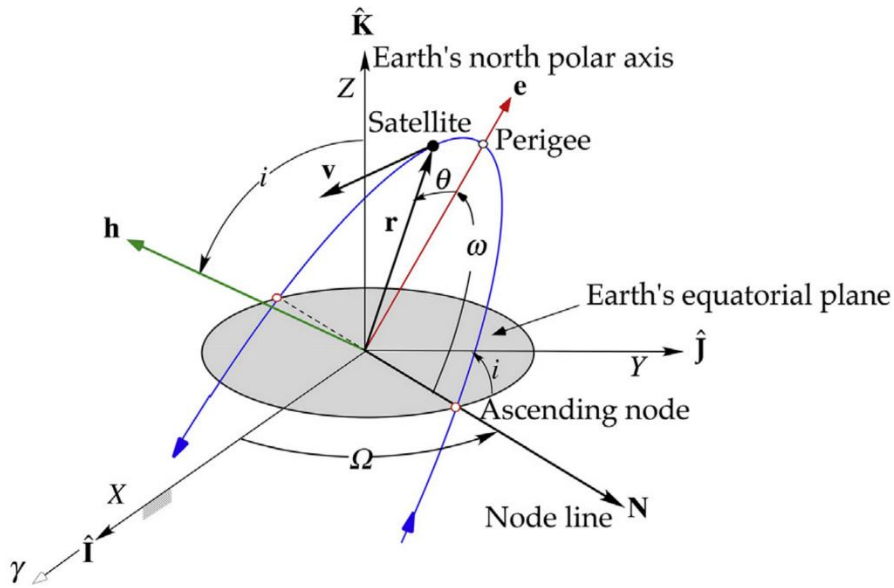


Figure 10 An orbit in 3D is defined by following parameters: $i, \theta,\ \Omega,\ \omega, \epsilon, h$ *[9]*

In summary, the six orbital elements are

- h: specific angular momentum.
- i: inclination.
- Ω: right ascension of the ascending node.
- e: eccentricity.
- ω: argument of perigee.
- θ: true anomaly.

The angular momentum h and true anomaly θ are frequently replaced by the semimajor axis a and the mean anomaly M, respectively. [9]

3.1.2 Angles-Only Preliminary Orbit Determination

If we have 3 observations of an orbiting body, we can determine its orbit using only that data. This is done by Gauss Method. Suppose we have the angle measurements of the tracked body at $t_1,\ t_2\ and\ t_3$ as in the figure below.

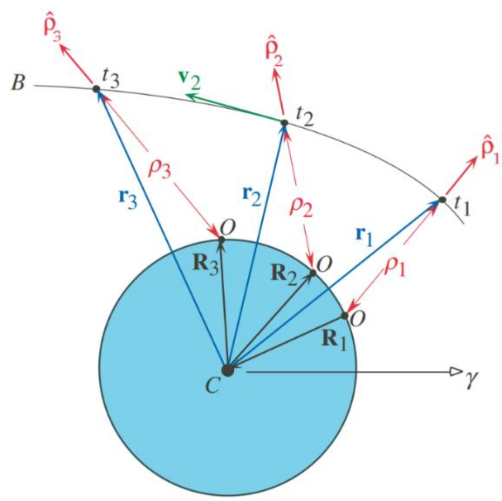


Figure 11 Center of attraction C, observer O, and tracked body B. [9]

The unit vectors $\hat{\rho}_1,\ \hat{\rho}_2,\ \hat{\rho}_3$ are determined using measuring the right ascension α and

declination δ of the body at each of the three times.

$$\boldsymbol{r_1} = \boldsymbol{R_1} + \rho_1\hat{\rho}_1$$

$$\boldsymbol{r_2} = \boldsymbol{R_2} + \rho_2\hat{\rho}_2$$

$$\boldsymbol{r_3} = \boldsymbol{R_3} + \rho_3\hat{\rho}_3$$

Since each r is 3-vector from these 9 equations we have 12 unknowns: each r is 3-vector and we need ρ_1, ρ_2, ρ_3.

3 more equations come from the fact that $\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}$ is on a plane since the angular momentum should be conserved.

$$\boldsymbol{r_2} = c_1\boldsymbol{r_1} + c_3\boldsymbol{r_3}$$

Adding $c_1\ and\ c_2$ we now have 12 equations and 14 unknowns.

Another consequence of the two-body equation of motion is that the state vectors r and v of the orbiting body can be expressed in terms of the state vectors at any given time by means of

the Lagrange coefficients. For the case at hand, this means we can express the position vectors r1 and r3 in terms of the position r2 and velocity v2 at the intermediate time t2 as follows [9]:

$$\boldsymbol{r_1} = f_1\boldsymbol{r_2} + g_1\boldsymbol{v_2}$$

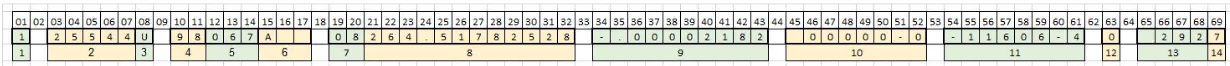
$$\boldsymbol{r_3} = f_3\boldsymbol{r_2} + g_3\boldsymbol{v_2}$$

Adding these equations now we have 18 equations and 18 unknowns, therefore it can be solved.

3.1.3 Two-Line Element Set

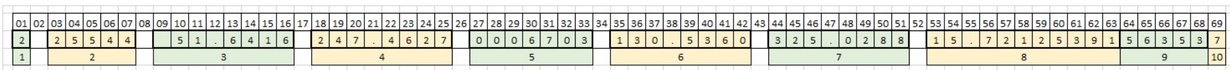
TLE is a data format of orbital parameters that are used to track objects such as satellites, debris, etc. in orbit. It consists of two lines which indicate a number of information including satellite name, orbital elements, epoch, etc.

LINE 1



Field	Columns	Content	Example
1	01–01	Line number	1
2	03–07	Satellite catalog number	25544
3	08–08	Classification (U=Unclassified, C=Classified, S=Secret) [11]	U
4	10–11	International Designator (last two digits of launch year)	98
5	12–14	International Designator (launch number of the year)	067
6	15–17	International Designator (piece of the launch)	A
7	19–20	Epoch Year (last two digits of year)	08
8	21–32	Epoch (day of the year and fractional portion of the day)	264.51782528
9	34–43	First Derivative of Mean Motion aka the Ballistic Coefficient [12]	-.00002182
10	45–52	Second Derivative of Mean Motion (decimal point assumed) [12]	00000-0
11	54–61	Drag Term aka Radiation Pressure Coefficient or BSTAR (decimal point assumed) [12]	-11606-4
12	63–63	Ephemeris type (internal use only - always zero in distributed TLE data) [13]	0
13	65–68	Element set number. Incremented when a new TLE is generated for this object. [12]	292
14	69–69	Checksum (modulo 10)	7

LINE 2



Field	Columns	Content	Example
1	01–01	Line number	2
2	03–07	Satellite Catalog number	25544
3	09–16	Inclination (degrees)	51.6416
4	18–25	Right Ascension of the Ascending Node (degrees)	247.4627
5	27–33	Eccentricity (decimal point assumed)	0006703
6	35–42	Argument of Perigee (degrees)	130.5360
7	44–51	Mean Anomaly (degrees)	325.0288
8	53–63	Mean Motion (revolutions per day)	15.72125391
9	64–68	Revolution number at epoch (revolutions)	56353
10	69–69	Checksum (modulo 10)	7

Figure 12 TLE data format and corresponding information [10]

We will use this format to track satellites in the project. Specifically, we take the past TLE information of a satellite and try to guess future states only using this data and then synthesize a new TLE using our algorithms.

3.2 Results

First, we gather data from the web. We use the past TLE data of RASAT and GOKTURK-2 satellites. After getting the data we visualize it over time.

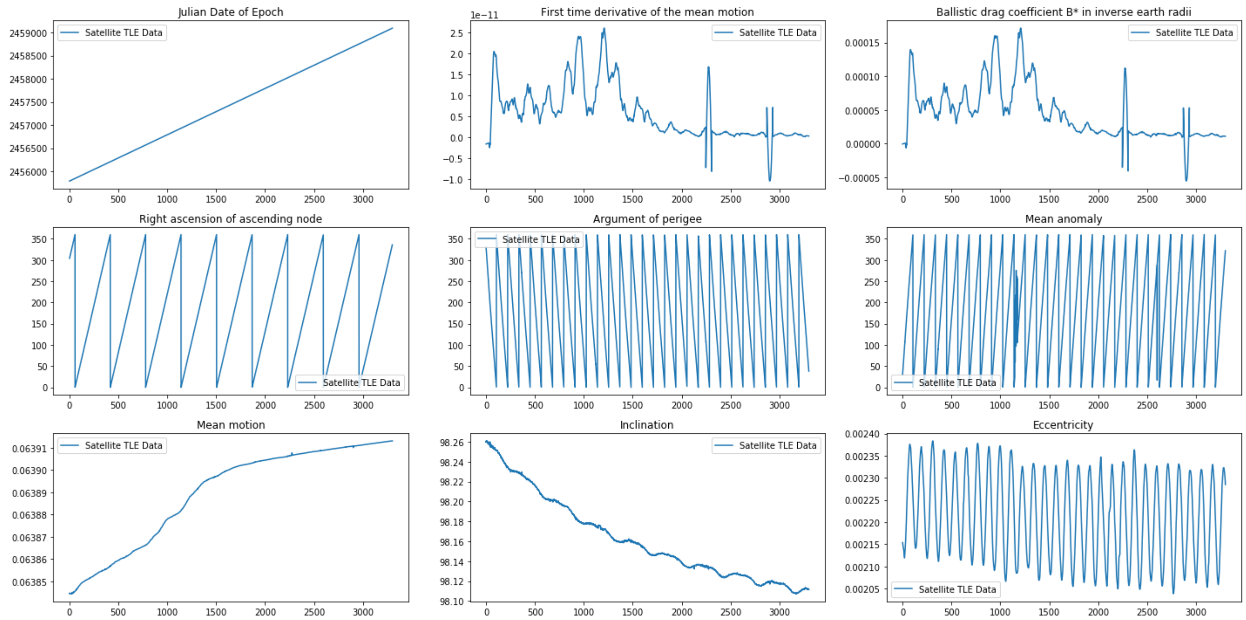


Figure 13 Gathered information from TLE data. RASAT

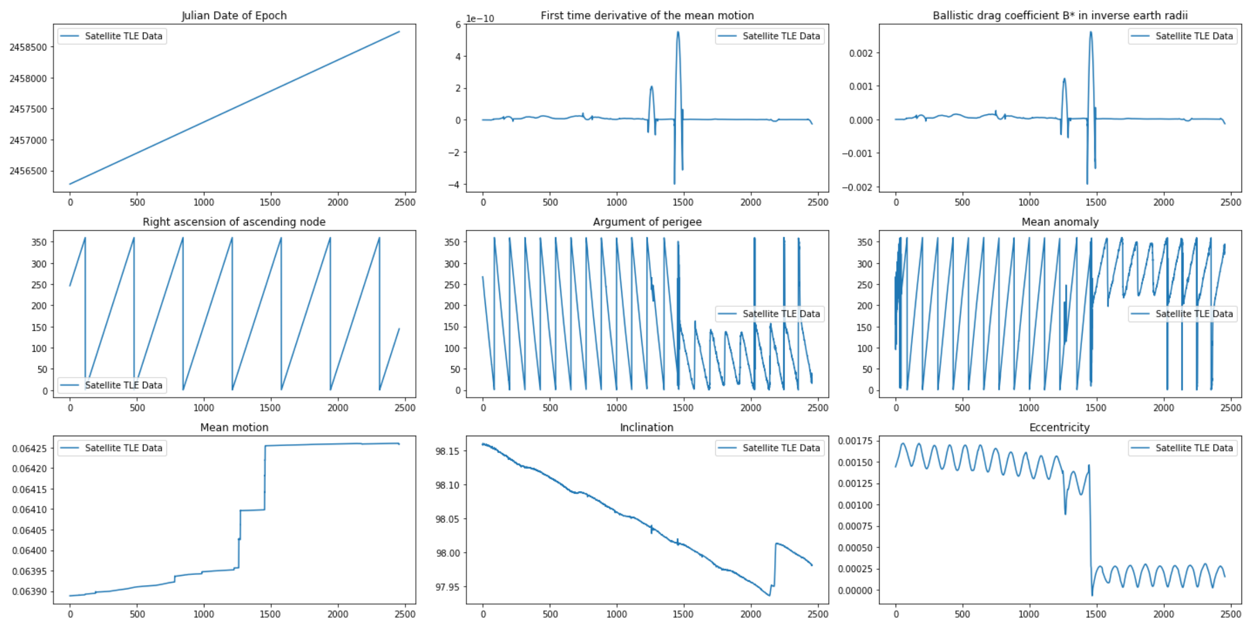


Figure 14 Gathered information from TLE data. GOKTURK-2

Then we implement a function to visualize the different positions of the satellite using SGP4 library.

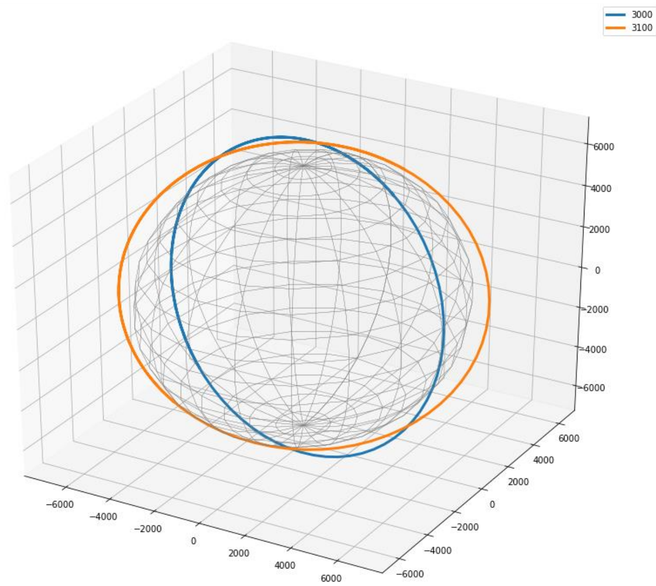


Figure 15 Different orbits of the satellite for different time points.

Then using the past TLE data we tried different fitting methods to best fit to data. First, we removed the wrong data from the dataset because some measurements were not correct. Afterward we tried polynomial fitting, sinusoidal fitting on the data. Some of them worked very well and fit the data and for some of them such as inclination, mean motion, etc. curve fitting methods don't work so we used the power of ML to fit data. We used LSTM layers since those are the most efficient way of fitting to time series. After hours of training, we got better results.

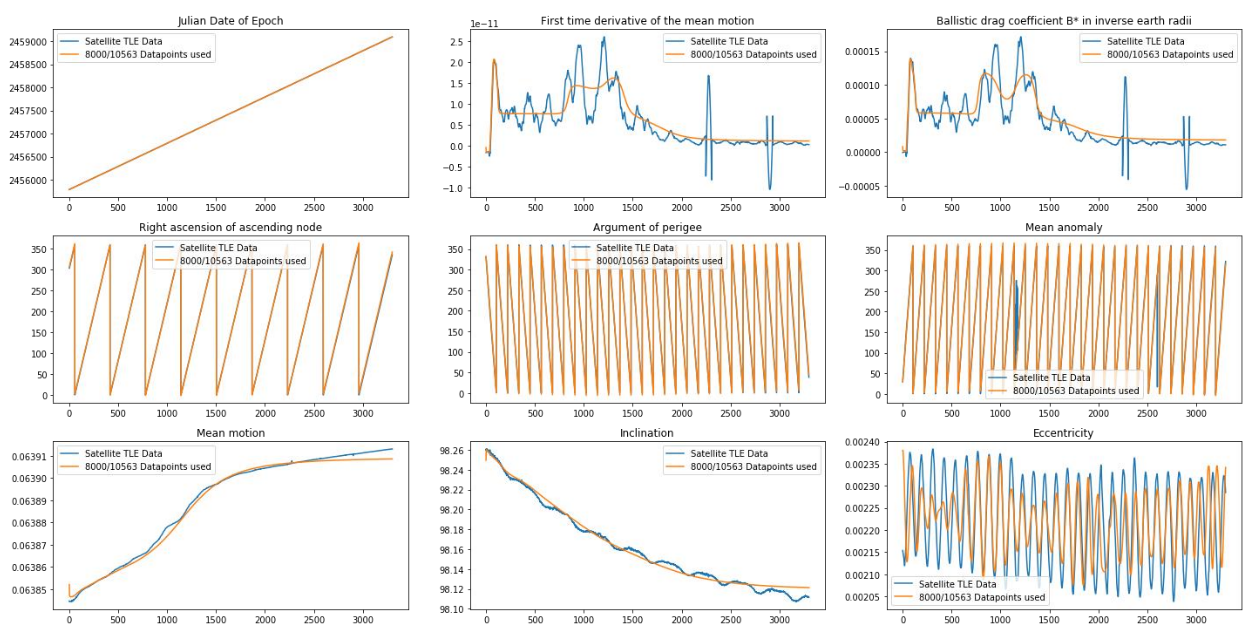


Figure 16 Fitting results. Only 8000 datapoints of the dataset are used to predict the next 2500 points

Before feeding the networks with data we normalized the data and dropout every 2/3 of data to decrease the computation. We used 8000 first data to guess the remaining 2500 datapoints. We can say that we achieved to get a meaningful result for each data.

Our next step was to implement a TLE synthesizer. The input of this function is the predicted data and it outputs TLE data. We further implemented a function to compare our results with the real data. So, this function gets two TLE data as input and a day and prints the maximum absolute distance between them on that day. We used again SGP4 algorithm to compute the coordinates of the satellite in the future using TLE data

Bibliography

[1] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/T%C3%9CB%C4%B0TAK_Space_Technologies_Research_Institute.

[2] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/RASAT.

[3] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/G%C3%B6kt%C3%BCrk-2.

[4] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/T%C3%BCrksat_6A.

[5] "TUBITAK UZAY," [Online]. Available: https://uzay.tubitak.gov.tr/en/projeler/imece.

[6] C. Langton, "Fundamentals of Signals".

[7] Anonymous, "Error Detection and Correction in Digital".

[8] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hamming_code#:~:text=In%20computer%20science%20and%20telecommunication,without%20detection%20of%20uncorrected%20errors.&text=For%20each%20integer%20r%20%E2%89%A5,2r%20%E2%88%92%20r%20%E2%88%92%201..

[9] H. D. Curtis, Orbital Mechanics for Engineering Students, Elsevier Ltd. , 2010.

[10] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Two-line_element_set.

[11] "Gezgin," [Online]. Available: https://gezgin.gov.tr.

[12] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Burst_error-correcting_code.