



**Politechnika Krakowska
im. Tadeusza Kościuszki**

Wydział Fizyki, Matematyki i Informatyki



Ewa Borsuk

Nr albumu: 104780

Analiza danych sondy Mars Express

Mars Express data analysis

**Praca magisterska
na kierunku FIZYKA TECHNICZNA**

Praca wykonana pod kierunkiem
dr Radosław Kycia
Instytut Fizyki

Uzgodniona ocena:

.....
podpisy promotora i recenzenta

Kraków 2018

Spis treści

0.1	Wstęp	5
0.2	Cel, zakres i metodyka pracy	5
1	Część teoretyczna	7
1.1	Sondy planetarne	8
1.2	Historia badań Marsa	11
1.3	Mars Express	14
1.4	Analityka danych	16
1.4.1	Python w analityce danych	16
1.4.2	Pakiety Pythona w analityce danych	17
1.5	Biblioteka astronomiczna de430	17
1.6	Kaggle	18
1.7	Mars Express Power Hackathon	19
2	Część praktyczna - Analiza danych	21
2.1	Opis danych	22
2.2	Plan analizy danych	22
2.3	Analiza kątów paneli słonecznych	22
2.3.1	Analiza danych pliku saaf	22
2.3.2	Analiza poprawności pomiaru kątów	26
2.4	Analiza odległości Słońce-Mars, Ziemia-Mars	29
2.5	Statystyka wydawania komend	33
2.6	Podsumowanie	34
2.7	Appendix	36
2.7.1	Kod analizy pliku evtf	36
2.7.2	Kod analizy pliku ltdata	38
2.7.3	Kod analizy pliku saaf	40

Abstrakt

W pracy przeanalizowano dane zebrane przez sondę Mars Express Orbiter. W tym celu użyto środowiska Jupiter, języka programowania Python, pakietów Pandas, Matplotlib i biblioteki astronomicznej NASA de430. Dane zostały udostępnione do domeny dzięki platformie Kaggle. Dane pochodzą z okresu w którym sonda doświadczała technicznych problemów. W pracy miałam na celu zaznajomienie się z charakterem danych. Podjęto próbę wyekstrahowania jak największej ilości informacji i ich sprawdzenia. Praca zawiera dane wychylenia kątów paneli słonecznych, podjęto próbę oszacowania czasu obiegu sondy dookoła Marsa. Sprawdzono jak dokładnie mierzona jest odległość Mars-Ziemia, Mars-Słońce porównując dane z tablicami astronomicznymi NASA, a także podano wytłumaczenie problemów w działaniu sondy, analizując procentowy udział wywoływania różnych podsystemów.

Abstract

In the thesis the analysis of data collected by the Mars Express Orbiter is presented. Data analysis were conducted in Jupyter environment, with Python programming language, packages Pandas and Matplotlib, and NASA astronomic database de430. Mars Express data used in these thesis were uploaded in public domain by Kaggle website. The data were taken during the failure of the probe. The aim of the thesis was to learn on the data structure. I presented the attempt to extract as many of useful information as it is possible and check them. The thesis contains the analysis of the degrees of deflection of solar panels of the probe, and the attempt to estimate time circulation around the Mars. It was checked if the distance Mars-Earth, Mars-Sun is measured precisely by comparing with astronomical tables of NASA, and its was tried to explain the wrong behaviour of the probe by analysis of percentage of calling sequence of the probe subsystems.

0.1 Wstęp

Ludzkość interesowała się obiektami na niebie od najdawniejszych czasów. Jednak dopiero rozwój technologii w XX i XXI wieku pozwolił na dokładniejsze badania, na początku przy użyciu teleskopów, a obecnie zdalnie, przy użyciu sond kosmicznych [2]. Badania te mają na celu poznanie dynamiki i warunków panujących na innych planetach. W obecnych czasach głównym celem badań, szczególnie sond i łazików jest Mars. Związane jest to z planami kolonizacji tej planety w niedalekiej przyszłości.

Jedną z misji w kierunku Czerwonej Planety jest Mars Express [21], która miała za cel zrobienie zdjęć powierzchni Marsa o wysokiej rozdzielczości, zbadanie składu atmosfery planety oraz jej cyrkulacji, sprawdzenie warstw podpowierzchniowych jak również ich wpływu na atmosferę Marsa [1]. Sonda taka zbiera wiele danych dotyczących telemetrii, zasilania i warunków panujących na planecie, którą bada. Część takich danych jest udostępniana do analizy dla każdego w Internecie [39]. Otwiera to możliwość samodzielnego poznania zasad, na których opiera się działanie sondy, a także możliwość wyciągnięcia wniosków o jakości przygotowania misji.

0.2 Cel, zakres i metodyka pracy

Celem tej pracy jest analiza i wizualizacja danych zebranych przez sondę kosmiczną Mars Express Orbiter [21] opublikowanych na portalu Kaggle [39]. Ważnym celem jest również nauka pracy z plikami sond kosmicznych i analizy big data. Praca jest podzielona na dwie części. Część I ma charakter teoretyczny, natomiast część II jest praktyczna i obejmuje analizę danych wraz z wizualizacją wyników.

W części I przedstawię krótki opis sond planetarnych i ich misji, historię odkryć na Marsie i misję sondy Mars Express. Następnie przejdę do zagadnień związanych z analityką danych. Skupię się głównie na języku Python i jego pakietach, ponieważ jest to obecnie jeden z najbardziej popularnych języków do analizy danych. W kolejnym kroku zaprezentuję bibliotekę astronomiczną de430, zaprojektowaną przez NASA do opisu ruchu planet Układu Słonecznego, Księżyca i Słońca w latach 1550–2650. Na koniec części I przedstawię platformę Kaggle służącą do udostępniania big data i nauki analizy danych, oraz omówię główne wyzwania Mars Express Power Hackaton zamieszczonego na stronie Kaggle. Dane udostępnione w domenie publicznej na potrzeby Mars Express Power Hackaton posłużyły do analizy w II części niniejszej pracy.

II część pracy skupia wyniki, które uzyskałam przy użyciu środowiska Jupyter, języka programowania Python, pakietów takich jak Pandas, Matplotlib, biblioteki astronomicznej de430. Dane uzyskane przez platformę Kaggle, dotyczące sondy Mars Express, przetworzyłam przy użyciu ww. narzędzi. W II części pracy przedstawię szczegółowy opis danych zamieszczonych na Kaggle, omówię plan analizy danych i zaprezentuję jej wyniki wraz z komentarzem.

W podsumowaniu przedstawię najważniejsze rezultaty analizy danych sondy Mars Express i zaprezentuję wnioski wynikające z analizy.

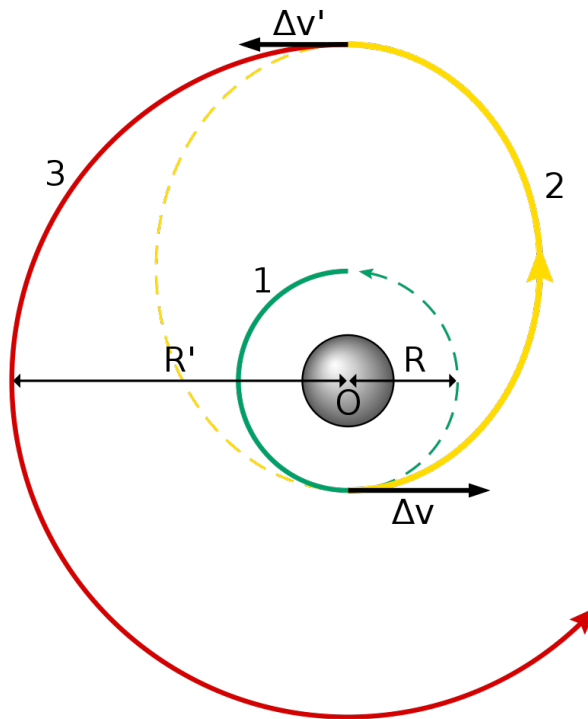
Rozdział 1

Część teoretyczna

1.1 Sondy planetarne

Sonda kosmiczna jest bezzałogowym statkiem kosmicznym, której celem jest przemierzanie kosmosu i jego badanie [2]. Sonda kosmiczna może przemierzać przestrzeń międzyplanetarną, orbitować wokół planet lub asteroid, jak również wylądować na nich bądź je minąć, aby polecieć w dalszą przestrzeń kosmiczną. Celem sond planetarnych są planety Układu Słonecznego. Agencje kosmiczne USA, Rosji, Japonii i Chin od drugiej połowy XX wieku wysyłały w kosmos sondy kosmiczne. W 2016 aktywnych było 15 misji [3].

Po przekroczeniu drugiej prędkości kosmicznej, sonda zaczyna orbitować wokół Słońca. Aby zmienić jej trajektorię i posłać w dalszą przestrzeń kosmiczną, najczęściej używa się manewru transferowego Hohmanna[5]. Polega on na dwukrotnym wykorzystaniu silników w celu wejścia sondy na wyższą lub niższą orbitę. Schemat tego manewru przedstawiony jest na Rysunku 1.1. Składa się on z trzech etapów. Rozpoczynając na orbicie początkowej swój lot, sonda przy pomocy dodatkowego impulsu silników nabiera znacznej prędkości, tak że trafia na tzw. orbitę transferową Hohmanna (trajektoria oznaczona nr 2 na Rysunku 1.1). Następnie po osiągnięciu pożądanej odległości R' od Słońca silniki sondy zostają włączone po raz drugi w celu uzyskania prędkości orbitalnej. W ten sposób sonda trafia na nową orbitę (trajektoria nr 3 na Rysunku 1.1).



Rysunek 1.1: Schemat manewru transferowego Hohmanna[6]

Kolejnym sposobem zmiany trajektorii sondy jest użycie tzw. asysty grawitacyjnej [7]. Polega ona na wykorzystaniu pola grawitacyjnego obiektu astronomicznego (np. planety), w celu zmiany szybkości i kierunku lotu sondy. Zaletą tej metody jest ograniczenie zużycia paliwa.

Innym sposobem jest podążanie za tzw. międzyplanetarną siecią transportu [8] (ang. Interplanetary Transport Network). Ta metoda polega na wykorzystaniu pola grawitacyjnego całego Układu Słonecznego.

go w celu wytyczenia drogi o najmniejszym zużyciu paliwa. Dużą wadą tej metody jest jej czasochłonność.

Tabela 1.1 przedstawia listę misji sond planetarnych, jakie zostały wysłane z Ziemi, w porządku chronologicznym. Celem każdej z tych sond były planety Układu Słonecznego lub inne obiekty astronomiczne zamknięte w obrębie Układu Słonecznego, takie jak asteroidy, komety, planety karłowate [9]. Jak do tej pory (2018 r.) wystartowały 42 takie sondy. W planach na najbliższe lata jest start sondy JUICE w 2022 roku i sondy Europa Clipper w latach 20-tych XXI w [10]. JUICE jest skrótem od JUpiter ICy moons Explorer. Zarówno JUICE jak i Europa Clipper będą zmierzać w kierunku Jowisza, w celu zbadania jego księżyców. JUICE będzie obserwował i zbierał dane na temat Ganimedesa (największego księżycy Jowisza), Kallisto (drugiego największego księżycy Jowisza) i Europy (czwartego największego księżycy Jowisza). Celem Europa Clipper jest sprawdzenie czy Europa posiada warunki korzystne dla życia [11]. NASA zdecydowała umieścić na Europa Clipper 9 instrumentów pomiarowych, m.in. magnetometr, który posłuży do sprawdzenia kierunku i siły pola magnetycznego Europy, co pozwoli naukowcom na obliczenie głębokości i zasolenia oceanu tego księżycy. W podobnym celu zostanie wysłany radar analizujący grubość pokrywy lodowej i szukający podpowierzchniowych jezior. Do potwierdzenia istnienia pod powierzchnią lodową oceanu posłuży również analiza grawitacyjna Europy.

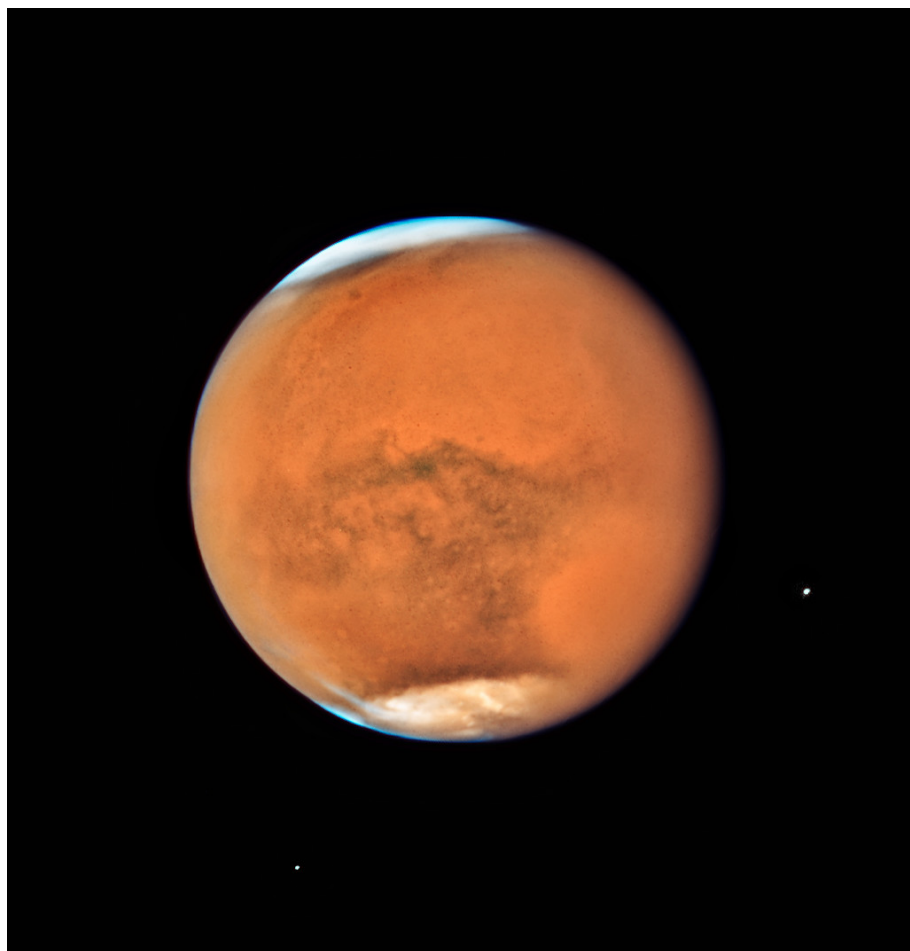
Pierwszą sondą, która dotarła do Merkurego, była Mariner 10. Wystartowała w 1973 roku. Pierwszą, która sfotografowała jądro komety, była Giotto z 1985 roku. Mariner 10 została wystrzelona przez NASA, natomiast Giotto była sondą ESA. Ta sama agencja wysłała w 2003 roku sondę Mars Express. W następnym rozdziale przyjrzymy się dokładniej historii badań Marsa i sondom, które do niego dotarły.

Nazwa sondy	Kraj/agencja	Data startu	Masa
Giotto	ESA	2 lipca 1985	583 kg
Suisei	Japonia	18 sierpnia 1995	140 kg
Phobos 1	ZSRR	7 lipca 1988	6220 kg
Phobos 2	ZSRR	12 lipca 1988	6220 kg
Magellan	NASA	4 maja 1989	3453 kg
Galileo	NASA	18 października 1989	2715 kg
Ulysses	NASA/ESA	6 października 1990	371 kg
Mars Observer	NASA	25 września 1992	2565 kg
NEAR	NASA	17 lutego 1996	805 kg
MGS	NASA	7 listopada 1996	1062 kg
Mars 96	RSA	16 listopada 1996	6180 kg
Mars Pathfinder	NASA	4 grudnia 1996	890 kg
Cassini	NASA	15 października 1997	5655 kg
Nozomi	JAXA	3 lipca 1998	540 kg
Deep Space 1	NASA	24 października 1998	486 kg
Mars Climate Orbiter	NASA	11 grudnia 1998	629 kg
Mars Polar Lander	NASA	3 stycznia 1999	583 kg
Stardust	NASA	7 lutego 1999	385 kg
Mars Odyssey	NASA	7 kwietnia 2001	725 kg
Genesis	NASA	8 sierpnia 2001	636 kg
CONTOUR	NASA	3 lipca 2002	775 kg
Hayabusa	JAXA	9 maja 2003	510 kg
Mars Express	ESA	2 czerwca 2003	1123 kg
Spirit	NASA	10 czerwca 2003	1063 kg
Opportunity	NASA	6 lipca 2003	1063 kg
Rosetta	ESA	2 marca 2004	3065 kg
MESSENGER	NASA	3 sierpnia 2004	1093 kg
Deep Impact	NASA	12 stycznia 2005	1010 kg
MRO	NASA	12 sierpnia 2005	2180 kg
Venus Express	ESA	9 listopada 2005	1240 kg
New Horizons	NASA	19 stycznia 2006	478 kg
Phoenix	NASA	4 sierpnia 2007	664 kg
Dawn	NASA	27 września 2007	1218 kg
Akatsuki	JAXA	20 maja 2010	500 kg
Juno	NASA	5 sierpnia 2011	3625 kg
Phobos-Grunt	RSA	8 listopada 2011	13 505 kg
Yinghuo-1	CNSA	8 listopada 2011	110 kg
Curiosity	NASA	25 listopada 2011	3893 kg
MOM	ISRO	5 listopada 2013	1337 kg
MAVEN	NASA	18 listopada 2013	2454 kg
Hayabusa 2	JAXA	3 grudnia 2014	590 kg
InSight	NASA/ESA	8-27 marca 2016	350 kg
JUICE	ESA	Czerwiec 2022	4900 kg
Europa Clipper	ESA	2021-2022	4500-6100 kg

Tabela 1.1: Sondy planetarne wg chronologii startu misji[9]

1.2 Historia badań Marsa

Mars jest czwartą planetą od Słońca. Jego masa wynosi około $6,4171 \cdot 10^{23}$ kg, natomiast jego promień jest równy około połowie promienia Ziemi [12]. Rysunek 1.2 przedstawia zdjęcie Marsa wykonane w 2018 roku przez kosmiczny teleskop Hubble'a. Charakterystyczny czerwony kolor planeta zawdzięcza tlenkom żelaza, którymi pokryta jest jej powierzchnia. Mars zalicza się do planet wewnętrznych Układu Słonecznego, posiada cienką warstwę atmosfery. W historii Marsa wielokrotnie miały miejsce kolizje z innymi ciałami niebieskimi, przez co jego powierzchnia jest pokryta wieloma kraterami, ma również wiele dolin, obszarów górzystych, czapy lodowe, jak również pustynie. Posiada największy kanion w Układzie Słonecznym o głębokości 5-8 km, długości 5000km i szerokości ok. 200km. Nosi nazwę Valles Marineris. Największa nizinna na Marsie to Vastitas Borealis, pokrywa aż 40% powierzchni planety. Najwyższą górą na Marsie jest Olympus Mons, która jest uznawana za najwyższą górę w Układzie Słonecznym, o wysokości 21 229 m. Jako, że na Marsie nie ma mórz i oceanów, wysokość wzniesień liczy się od poziomu w którym obliczono ciśnienie atmosferyczne 6,105 hPa.



Rysunek 1.2: Zdjęcie Marsa wykonane przez kosmiczny teleskop Hubble'a w 2018 roku [13]

Obecnie o Marsie i jego topografii wiemy dość dużo. Jednak wszystkie informacje, które obecnie posiadamy, były zbierane przez wiele setek lat obserwacji i badań. Pierwsze obserwacje Marsa były prowadzone przez Babilończyków, starożytnych Egipcjan, Greków i Rzymian [14]. Przed wynalezieniem teleskopu w XVII wieku, jedynymi obserwacjami, jakie mogli przeprowadzić starożytni badacze, był intensywny czerwony blask Marsa na nocnym niebie i jego charakterystyczny sposób poruszania się po nieboskłonie, niepodobny do żadnego innego obiektu astronomicznego. Starożytni Babilończycy wykonywali obserwacje astronomiczne już w V wieku p.n.e.

Pierwszymi przełomowymi pomiarami były obserwacje wykonane przez Tycho Brahe, duńskiego astronoma [15]. W 1576 założył on obserwatorium astronomiczne na wyspie Hven, niedaleko Kopenhagi. Przez następne 20 lat badał niebo i prowadził badania wyłącznie przy użyciu własnego wzroku i ogromnych instrumentów pomiarowych. Teleskop był jeszcze wtedy nieznan. Co ciekawe, jego pomiary były niezwykle dokładne. Na ich podstawie, był w stanie obliczyć pozycję Marsa z dokładnością do 4 minut kątowych.

Kolejnym gigantycznym skokiem w rozumieniu trajektorii ruchu Marsa i jego topografii, był XVII wiek. To właśnie na początku XVII wieku, w roku 1609, Johannes Kepler, uczeń Tycho Brahe, opublikował "Astronomia nova". W swoim dziele, Kepler przedstawił swoje dwa pierwsze prawa i zaproponował, że orbitą Marsa jest elipsa. Było to o tyle niezwykle założenie, że w tamtej epoce powszechnie uważano okrąg za kształt idealny i zakładano, że wszystkie orbity są okrągłe [16]. W tym samym roku Galileo Galilei skonstruował pierwszy, prymitywny teleskop, przy użyciu którego mógł przyrzeć się m.in. Marsowi. W kolejnych latach badania Marsa posuwały się na przód. W 1659 roku, Christiaan Huygens sporządził dokładne rysunki Marsa dzięki użyciu zaawansowanego, skonstruowanego przez siebie teleskopu. W roku 1666, Giovanni Cassini określił długość dnia na Marsie. Pod koniec XVII wieku Christiaan Huygens znów zapisał się na kartach historii astronomii. W 1672 roku zaobserwował białą plamę w okolicach bieguna południowego Marsa. W 1698 roku opublikował "Cosmotheros", pracę, w której spekuluje na temat życia pozaziemskiego i warunków jakie powinny być spełnione, by takie życie powstało. Jest to pierwsza taka publikacja w dziejach ludzkości.

W wieku XVIII uwagę naukowców skupiły "białe plamy" na biegunie południowym Marsa. W 1704 roku Giancomo Miraldi odkrył, że centrum plam nie pokrywa się z rotacyjnym biegunem [17]. W 1719 roku Miraldi wysunął hipotezę, że "białe plamy" są tak naprawdę polarne czapy lodowe Marsa. W tym samym roku, Mars był w opozycji i znajdował się bardzo blisko Ziemi. Następnym razem tak blisko był dopiero w 2003 roku. Wywołało to zbiorową panikę. Mówiąc o historii odkryć związanych z Marsiem w XVIII, warto wspomnieć również o Sir Williamie Herschel. Używając własnoręcznie zbudowanego teleskopu, przeprowadził szereg obserwacji Marsa. Na ich podstawie wysunął jednak fałszywe wnioski, jakoby ciemne plamy na Marsie miały być oceanami, zaś jasne lądem. Uważał, że wszystkie planety są zamieszkałe przez inteligentne istoty pozaziemskie. Udało mu się określić dość dokładnie kąt nachylenia osi obrotu Marsa do normalnej orbity okołosłonecznej. Określił go jako 30° , podczas gdy współczesne badania podają $25,19^\circ$.

Wiek XIX przyniósł wydarzenia, które później zostały określone mianem "kryzysu kanałów Marsjańskich". W 1854 roku William Whewell stwierdził, że Mars posiada zielone morza i czerwone lądy [18]. W 1867 Richard Anthony Proctor opublikował mapę Marsa zawierającą opis domniemanych mórz i kontynentów. W 1877 roku, Giovanni Schiaparelli przedstawił propozycję nomenklatury dla obiektów topograficznych na Marsie. Do opisu prostych, przecinających się linii na powierzchni Marsa, użył włoskiego terminu "canali", który później został błędnie przetłumaczony jako kanał. Do opinii publicznej przedostała się zatem informacja, że naukowcy odkryli kanały wybudowane prawdopodobnie przez pozaziemską cywilizację. Dopiero w roku 1894, Percival Lowell po przeprowadzeniu wnikliwych obserwacji w obserwatorium w Flagstaff, w Arizonie, zdementował istnienie kanałów na Marsie i uznał je za złudzenie optyczne.

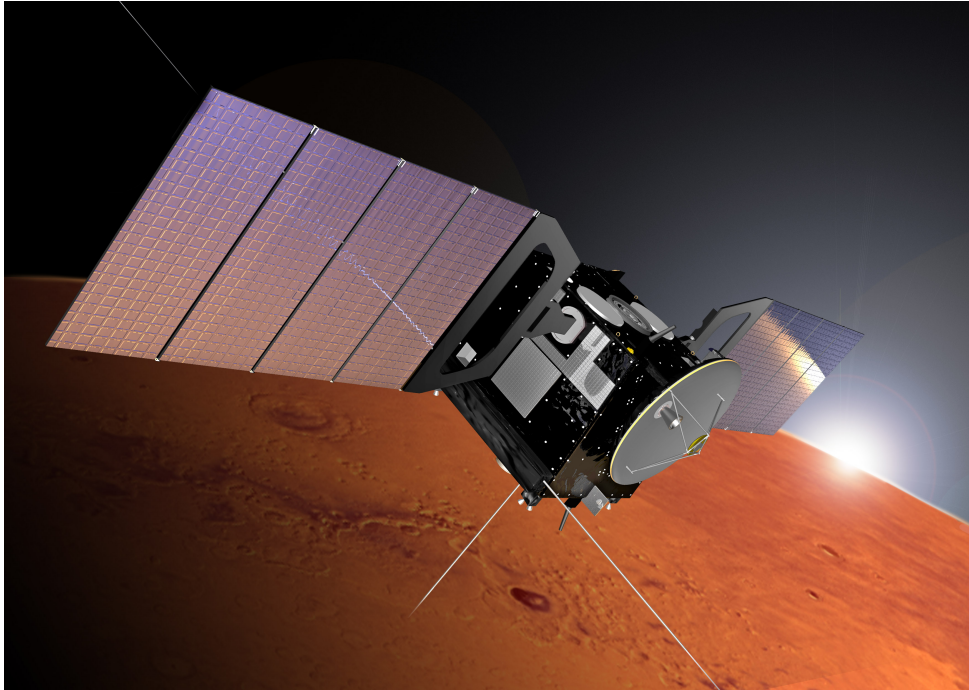
W wieku XX i na początku XXI technologia rozwinęła się bardzo dynamicznie, dzięki czemu możliwe było tak dokładne zbadanie Marsa jak nigdy w historii. Począwszy od przelotów obok planety, poprzez orbiter, aż do lądowców i łazików, technologia kosmiczna pozwoliła nam bardzo wnikliwie poznać topo-

logię, atmosferę i skład Marsa. Do najwcześniejszych lotów w okolicach Marsa, które przyniosły pierwsze zdjęcia Marsa z przestrzeni kosmicznej, należały Mariner 3-4 i Mariner 6-7 [19]. Misjami, których celem było utrzymanie sondy na orbicie wokół Marsa, były Mariner 8-9, Viking 1-2, Mars Observer, Mars Global Surveyor, Mars Climate Orbiter, 2001 Mars Odyssey, Mars Express, Mars Reconnaissance Orbiter, Mars Atmosphere and Volatile Evolution, NASA In ESA's ExoMars Orbiter. Dzięki tym misjom możliwe było przeprowadzenie długotrwałych badań Marsa, w skali całego globu. Do najbardziej skomplikowanych misji należą misje lądowników i łazików. Jak do tej pory udało się przeprowadzić następujące misje na powierzchni Marsa: Viking 1-2, Pathfinder, Polar Lander/Deep Space 2, Mars Exploration Rovers, Phoenix, Mars Science Laboratory, InSight, NASA In ESA's ExoMars Rover. W planach NASA jest start misji wraz z lądownikiem w 2020 roku. Do bardziej futurystycznych planów NASA należą misje z użyciem samolotów i balonów na Marsie, misji podziemnych i transportu próbek z Marsa na Ziemię.

W następnym rozdziale przedstawię szczegółowo jedną z misji kosmicznych na Marsa, sondę Mars Express.

1.3 Mars Express

Sonda Mars Express [21] jest bezzałogową sondą kosmiczną, wysłaną przez Europejską Agencję Kosmiczną (ang. European Space Agency, skrót ESA). Jest to pierwsza misja badająca Marsa zorganizowana przez tę agencję. Sonda składa się z orbitera Mars Express Orbiter oraz lądownika Beagle 2. Na rysunku 1.3 przedstawiona jest wizualizacja sondy Mars Express udostępniona przez agencję ESA[20].



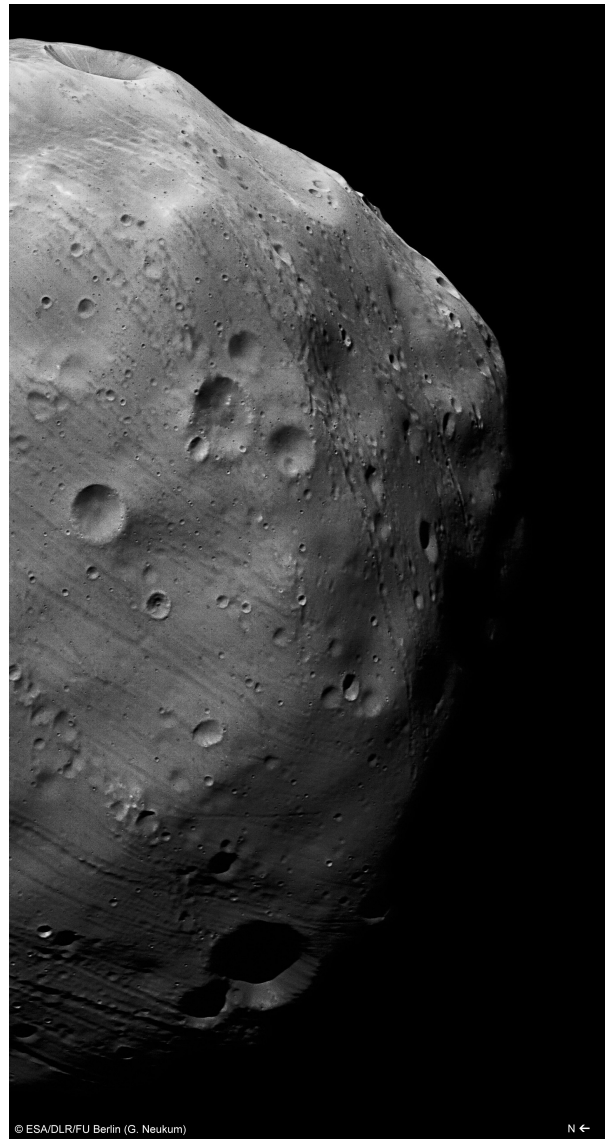
Rysunek 1.3: Sonda Mars Express[20]

Głównymi zadaniami Mars Express Orbiter było zrobienie zdjęć powierzchni Marsa o wysokiej rozdzielczości, zbadanie składu atmosfery planety oraz jej cyrkulacji, sprawdzenie warstw podpowierzchniowych jak również ich wpływu na atmosferę Marsa[21].

Z kolei lądownik Beagle 2 miał przeprowadzić badania geologiczne jak i atmosferyczne na powierzchni planety, jednak 25 grudnia 2013 utracono z nim kontakt podczas nieudanej próby lądowania. W 2015 roku sonda Mars Reconnaissance Orbiter wysłana przez NASA przesłała zdjęcia, na których widać lądownik Beagle 2. Ze zdjęć wynika, że przyczyną nieudanej misji Beagle 2 mogły być jedynie częściowo rozchylone panele słoneczne[22].

Sonda Mars Express oprócz Beagle 2 została wyposażona w 11 urządzeń badawczych. Po stracie lądownika Beagle 2, sonda przystąpiła do badań dwóch księżyców Marsa, Fobosa i Deimosa. Żadna wcześniejsza misja nie wykonała tak dokładnych zdjęć Fobosa[23]. Rysunek 1.4 przedstawia pierwsze zdjęcie Fobosa, jakie udało się wykonać sondzie Mars Express. Zostało wykonane 7 marca 2010 roku. Dzięki wyraźnie eliptycznej orbicie okołobiegunowej, sonda Mars Express co pięć miesięcy przybliża się do Fobosa. Zdjęcie z rysunku 1.4 zostało wykonane z rozdzielczością 4,4 metra na pixel. Podobnie jak księżyc orbitujący wokół Ziemi, Fobos ukazuje tylko jedną stronę w kierunku swojej planety. Aby zobaczyć jego ciemną stronę, należy wylecieć poza jego orbitę. Sondzie Mars Express udało się to zadanie 7, 10 i 13 marca 2010

roku. Podczas tej operacji sonda wykonała również kilka pomiarów Fobosa, przy użyciu instrumentów badawczych, w jakie została wyposażona. Fobos ma nieregularny kształt i wymiary $27 \times 22 \times 19$ km. Jego pochodzenie jest do tej pory niejasne. Część naukowców uważa, że jest pochwyconą przez Marsa planetoidą klasy C. Inni badacze skłaniają się ku hipotezie, że Fobos został uformowany wokół Marsa w czasie formowania się planety [25] [26].



Rysunek 1.4: Pierwsze zdjęcie Fobosa wykonane przez sondę Mars Express 7 marca 2010 roku [24]

Oprócz badania księżyców sonda Mars Express zajmowała się oczywiście badaniem również samego Marsa. Jednym z kluczowych zadań sondy było sprawdzenie czy na Marsie występowały kiedyś warunki umożliwiające powstanie i rozwój życia, oraz wykrycie obecności wody. Do najbardziej istotnych odkryć

sondy można zaliczyć odnalezienie minerałów, które formują się tylko w obecności wody, wykrycie podziemnych skupisk lodu oraz dowodów na to, że Mars jeszcze niedawno był aktywny wulkanicznie. Dzięki misji możliwe było stworzenie mapy chemicznej atmosfery Marsa, która zakłada możliwość występowania metanu- gazu powstającego na Ziemi w wyniku erupcji wulkanicznych i procesów biochemicznych.

1.4 Analityka danych

Analityka danych (ang. Data science) jest obecnie popularnym tematem zarówno w środowisku nauki jak i w biznesie. Nauka o danych obejmuje zagadnienia takie jak algebra liniowa, modelowanie statystyczne, wizualizacja, lingwistyka komputerowa, analizę grafów, uczenie maszynowe, analizę biznesową oraz przetwarzanie i pobieranie danych[27]. Jest to zatem temat bardzo szeroki i interdyscyplinarny. Analizę danych można przeprowadzać w środowiskach takich jak język R lub MATLAB, jednak w ostatnich latach coraz większą popularność zdobywa Python. W następnym rozdziale przedstawię główne powody, dla których Python stał się kluczowym językiem programowania w analizie danych.

1.4.1 Python w analityce danych

Python powstał w roku 1991. Jest językiem wysokiego poziomu, interpretowanym, obiektowym, służącym do ogólnego użytku. Właśnie jego wszechstronność jest głównym atutem, który przyciąga coraz więcej naukowców do używania go w swoich badaniach. Pozwala na przeprowadzanie szybkich eksperymentów, sprawdzenie teorii i łatwe wprowadzenie aplikacji naukowych. Ponadto Python posiada wiele innych cech, które sprawiają, że jest niezastąpiony w analityce danych:

- posiada wiele rozbudowanych pakietów do analizy danych i uczenia maszynowego, dzięki czemu wszystkie niezbędne narzędzia do analizy danych są łatwo dostępne.
- pozwala na łączenie różnych narzędzi, języków programowania, strategii przetwarzania danych i algorytmów uczenia. Posiada pakiety umożliwiające wywoływanie kodu w innych językach (m.in. Javie, C, Fortranie, R, Julii), jak również wysyłanie części obliczeń do tych języków, co podnosi wydajność skryptów.
- pozwala na zastosowanie wielu stylów programowania (obektowego, proceduralnego, funkcyjnego).
- działa bez problemów na różnych systemach operacyjnych. Programy napisane w Pythonie tak samo dobrze działają na Windowsie, Linuxie, jak i Mac OS. Nawet tak małe dystrybucje jak Raspberry Pi czy Arduino nie mają problemów z przetwarzaniem kodu Pythona.
- na tle innych języków interpretowanych (R, MATLAB-a), Python działa bardzo szybko. Jest co prawda wolniejszy niż języki takie jak C, Java, Julia, ale przy użyciu odpowiedniego kompilatora (np. Cythona, PyPy), można zwiększyć jego wydajność.
- potrafi doskonale zarządzać pamięcią, tak aby obciążać ją w minimalnym stopniu. Ułatwia to pracę na dużych zbiorach danych. Jeśli np. potrzebne jest wczytanie, przekształcenie, dzielenie, zapisanie i usunięcie danych w wielu krokach przetwarzania, mechanizm odzyskania pamięci przez Pythona jest bardzo użyteczny.
- Python jest bardzo prosty w nauce i użytkowaniu.

Największym atutem języka Python jest ogromna liczba bibliotek. Poniżej przedstawiam te, które zostaną wykorzystane do dalszej analizy.

1.4.2 Pakiety Pythona w analityce danych

Podstawą użytkowania Pythona jest pobranie odpowiednich pakietów. W analizie danych podstawowymi pakietami są:

- NumPy[28]- został opracowany przez Trávisa Oliphanta. Udostępnia użytkownikom wielowymiarowe tablice, jak również duży zestaw funkcji wykonujących rozmaite operacje matematyczne na tablicach. Aby zainstalować pakiet NumPy należy użyć komendy:
pip install numpy
- SciPy[29]- został stworzony przez Trávisa Oliphanta, Pearu Petersona i Erica Jonesa. Posiada on większą w stosunku do NumPy ilość funkcji matematycznych i algorytmów naukowych, pozwalających m.in. na użycie algebry liniowej i macierzy rzadkich, przetwarzanie sygnałów i obrazów, jak również optymalizacji danych i szybkiej transformaty Fouriera. Aby zainstalować pakiet SciPy należy użyć komendy:
pip install scipy
- Pandas[30]- został stworzony przez Wesa McKinney'a. Umożliwia wykonywanie operacji, które są niedostępne dla pakietów Numpy i Scipy. Dzięki zastosowaniu specjalnych struktur danych takich jak *DataFrame* i *Series*, możliwe jest zarządzanie dużymi tablicami złożonymi z danych różnych typów. Dzięki Pandas można również tworzyć szeregi czasowe. Pakiet ułatwia wczytywanie danych z różnych źródeł i pomaga w uzupełnianiu luk w danych, ich agregacji, przekształcaniu i wizualizacji. Aby zainstalować pakiet Pandas należy użyć komendy:
pip install pandas
- Matplotlib[31]- został stworzony przez Johna Huntera. Jest to biblioteka służąca do tworzenia wykresów na podstawie tablic, jak również interaktywnego wizualizowania danych. Aby zainstalować Matplotlib należy użyć polecenia:
pip install matplotlib
- Jupyter[32]- został stworzony przez Fernando Pereza. Jego interfejs pozwala na szybkie sprawdzanie hipotez poprzez powtarzalne eksperymenty. Jest powłoką obsługującą kilka języków programowania, pozwalającą na obliczenia równoległe i posiadającą modyfikowalne instrukcje. Aby zainstalować Jupytera należy użyć polecenia:
pip install jupyter

W kolejnym kroku omówię bibliotekę danych astronomicznych NASA, która pozwoli porównać informacje z sondy na temat odległości Mars-Ziemia z rzeczywistymi odległościami.

1.5 Biblioteka astronomiczna de430

Biblioteka astronomiczna de430 jest biblioteką stworzoną przez NASA opisującą efemerydy planet Układu Słonecznego, Księżyca i Słońca w latach 1550–2650[33]. Ta baza danych działa bardzo dobrze z pakietem Jet Propulsion Laboratory Ephemeris (w skrócie jplephem) i NumPy. Aby zainstalować jplephem wystarczy użyć komendy:

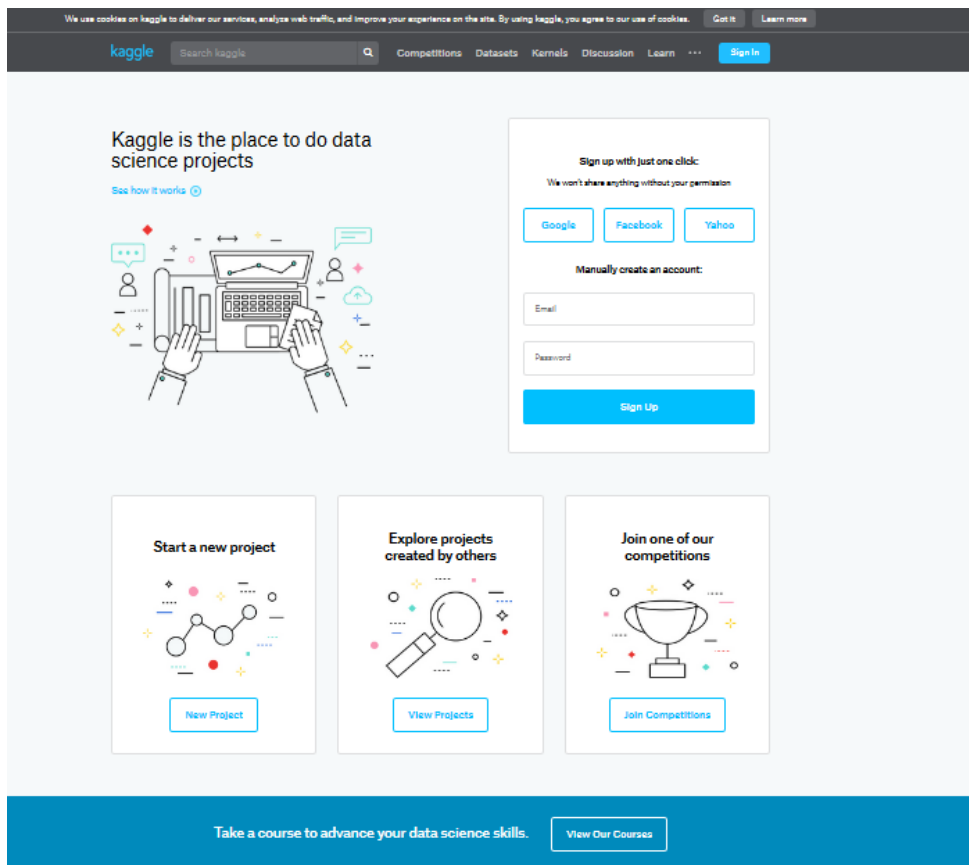
```
pip install jplephem
```

Biblioteka de430 została zaprojektowana przez Navigation and Ancillary Information Facility w NASA[34] (skrót NAIF). W ramach pracy NAIF powstały bazy danych opisujące efemerydy asteroid, komet, satelit i stacji kosmicznych[35]. Wśród bibliotek NAIF obejmujących efemerydy planet można wyróżnić de430,

de431, de432, de435[36]. Biblioteki de430 i de431 są do siebie bardzo podobne, różnią się jedynie podejściem w opisie dynamicznym ziemskiego Księżyca[37]. Biblioteka de430 używa bardziej dokładnego modelu ruchu Księżyca, dzięki czemu otrzymuje lepsze wyniki efemerydy dla położenia Księżyca dla obecnej epoki, jednak dla odległej przeszłości lepiej użyć biblioteki de431.

1.6 Kaggle

Kaggle jest platformą zajmującą się analityką danych i uczeniem maszynowym[39]. Powstała w kwietniu 2010 roku. Umożliwia udostępnianie baz danych, jak również autorskiego kodu z zakresu analityki danych. Na stronie można znaleźć konkursy na najlepsze modele przewidywania i opisu danych udostępnianych przez różnego rodzaju firmy, użytkowników i instytuty badawcze. Na rysunku 1.5 jest przedstawiona witryna strony. Korzystanie z danych, jak również rejestracja są darmowe. Na stronie są dostępne tutoriale dla osób początkujących i zaawansowanych. Kaggle oferuje również swoim użytkownikom wsparcie merytoryczne i pomoc w wykonaniu projektów. Przy pomocy Kaggle możliwe jest nauczanie się obsługi Pythona, uczenia maszynowego, Pandas, wizualizacji danych, SQL i R[40]. Takie podejście sprawia, że powstają nowe rozwiązania w wyniku kooperacji grup programistów z całego świata. W marcu 2017 roku Kaggle nawiązał współpracę z Google Cloud, pozostaje jednak osobną firmą.



Rysunek 1.5: Witryna Kaggle

1.7 Mars Express Power Hackathon

W 2017 roku na platformie Kaggle został zamieszczony konkurs na analizę danych sondy Mars Express Orbiter[41]. Niniejsza praca jest oparta o dane udostępnione na potrzeby tego hackathonu. Głównym zadaniem konkursowym było przedstawienie planu zarządzania temperaturą sondy na najbliższe 6 miesięcy, na podstawie danych udostępnionych z poprzedniego roku ziemskiego. Sonda pobiera energię elektryczną dzięki panelom słonecznym, a w czasie zaćmień z energii nagromadzonej w bateriach. Energia elektryczna jest zużywana nie tylko przez jednostki mechaniczne sondy, ale również przez podsystem utrzymujący sondę we wcześniej zadanym zakresie temperatur. W skład tego podsystemu wchodzi m.in. grzejnik oraz chłodziarka. Dopiero energia pozostała w nadmiarze w stosunku do zapotrzebowania układu sterującego sondy i systemu zarządzającego temperaturą, może zostać wykorzystana do celów badawczych. Dlatego właśnie tak kluczowym dla twórców konkursu było przedstawienie rozwiązania optymalnego zarządzania systemem kontrolującym temperaturę sondy. Niniejsza praca przedstawi autorską analizę danych zebranych przez Mars Express Orbiter wraz z komentarzem.

Rozdział 2

Część praktyczna - Analiza danych

2.1 Opis danych

W zestawie danych opublikowanych na platformie Kaggle w ramach Mars Express Power Hackathon, znajdują się następujące pliki:

- plik `dmop` -opisuje komendy wykonywane przez podsystemy sondy wraz z czasem wykonania,
- plik `evtf` -podobnie jak plik `dmop` opisuje wydarzenia i komendy wykonane przed podsystemy sondy wraz z czasem wykonania,
- plik `ftl` -opisuje komendy sterujące nawigacją sondy,
- plik `ltdata` -opisuje dynamikę lotu, np. odległość Słońce-Mars, Ziemia-Mars,
- plik `saaf` -opisuje położenie sondy względem Słońca i Marsa przy pomocy kątów.

Wszystkie pliki mają format CSV (ang. Comma Separated Values) - wartości rozseparowane przecinkami. Jest to format charakterystyczny dla prostych danych często stosowany w analityce danych.

2.2 Plan analizy danych

Dalsza część pracy będzie się składać z następujących elementów:

- opiszę zawartość pliku `saaf` i przeprowadzę analizę kątów paneli słonecznych i obliczę niepewności pomiarowe zestawu danych pliku `saaf`.
- analizując plik `ltdata`, przedstawię wykresy odległości Marsa od Słońca, Ziemi od Marsa i porównam je z danymi przekazanymi przez NASA w bibliotece `de430`.
- używając danych z pliku `evtf`, przedstawię statystykę wydawania komend przez podsystemy sondy.

Wszystkie wyniki przedstawione poniżej powstały w wyniku działania samodzielnie napisanych programów w języku Python. Programy te zostały zamieszczone w Appendixach na końcu pracy.

2.3 Analiza kątów paneli słonecznych

W tej sekcji zostanie przedstawiona analiza danych kątów wychylenia paneli słonecznych sondy. Zostanie oceniony stopień dokładności ich pomiaru, a także wyniki, które można z nich otrzymać, m.in. czas obiegu sondy wokół Marsa.

2.3.1 Analiza danych pliku `saaf`

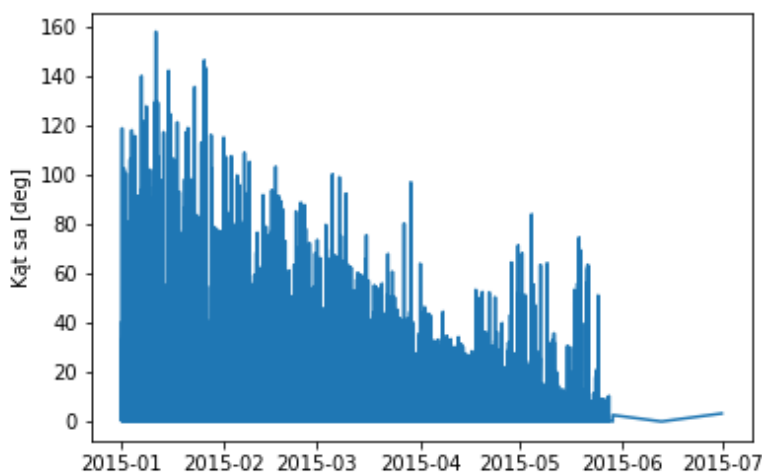
Tabela 2.1 opisuje podstawowe właściwości pliku `saaf`. Plik ten zawiera 301037 kątów sa , sx , sy , sz . Kąt sa to kąt między normalną do paneli słonecznych sondy a Słońcem, zaś kąty sx , sy , sz to odpowiednio kąty między osiami x , y , z sondy a linią Mars Express-Słońce. Kąty są podane w stopniach.

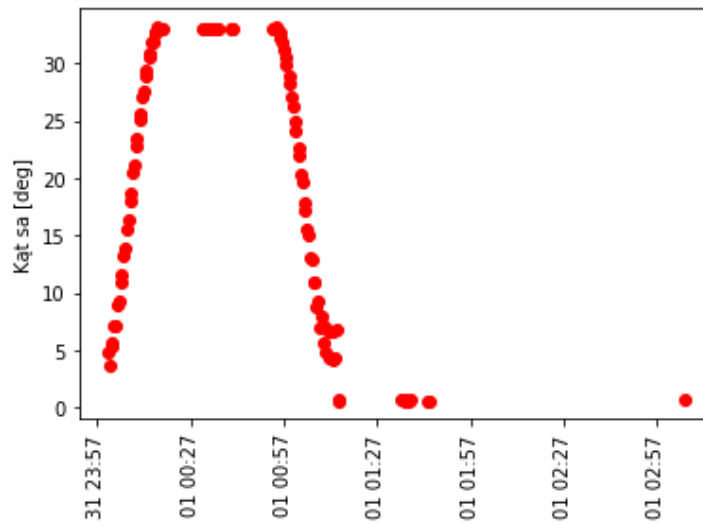
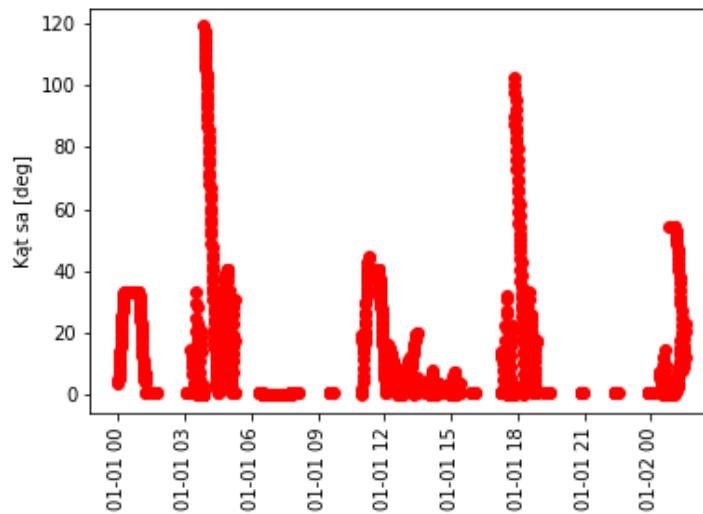
Wartość minimalna kąta sa to $0,00^\circ$, zaś maksymalna to $158,38^\circ$. Wartość średnia sa wynosi $6,28^\circ$. Rysunek 2.1 przedstawia zależność wartości kąta sa od czasu. Na wykresie widać dużo zaburzeń i silną tendencję wygaszania wartości kąta sa w kierunku $0,0^\circ$. Rysunek 2.2 przedstawia zbliżenie na wycinek wykresu z rysunku 2.1. Na osi OX rysunku 2.2 przedstawione są pierwsze 100 wartości kąta sa zawarte w pliku `saaf`, z okresu od 31 grudnia, godz. 23:57 do 1 stycznia, godz. 3:27. Ta pierwsza porcja danych wygląda dość obiecująco- wykres nie zawiera żadnych artefaktów ani zaburzeń. Na wykresie z rysunku 2.2

widać wyraźne, gładkie maksimum ze stopniowym wzrostem do wartości $33,23^\circ$. Obszar stabilnej wartości kąta sa przy $33,23^\circ$ prawdopodobnie oznacza optymalne dopasowanie kąta sa do Słońca, zaś obszar po godzinie 1:27, gdy wartość sa spada do zera, to prawdopodobnie czas zasłonięcia Słońca przez Marsa. W tym czasie panele słoneczne nie poruszały się i sonda była zasilana za pomocą baterii. Spodziewanym rezultatem jest odnalezienie podobnych picków w równych odstępach czasu, zgodnych z ruchem sondy wokół Marsa. Rysunek 2.3 przedstawia wykres zależności pierwszych 1200 wartości kąta sa od czasu, dzięki któremu można oszacować czas obiegu sondy dookoła Marsa. Widać na nim trzy podobne do siebie picki o wartościach $33,32^\circ$, $44,48^\circ$ i $54,50^\circ$. Czas wystąpienia maksymalnych wartości tych struktur to odpowiednio: godzina 00:16:32, 1 stycznia 2015 roku, 11:16:32 1 stycznia 2015 roku i 00:48:51 2 stycznia 2015 roku. Czas obiegu sondy wokół Marsa to zatem 11 godzin w pierwszym przypadku i 13 godzin 32 minuty w drugim. Dodatkowo wraz z upływem czasu różnice w kolejnych pomiarach stają się bardziej skokowe i zamiast stopniowego wzrostu wartości kąta sa jak na rysunku 2.2 można zaobserwować gwałtowne zmiany wartości sa . Takie zachowanie sondy może być spowodowane kłopotami z układem sterującym.

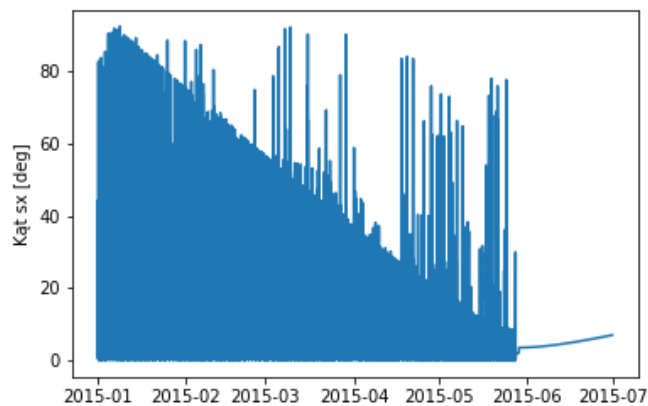
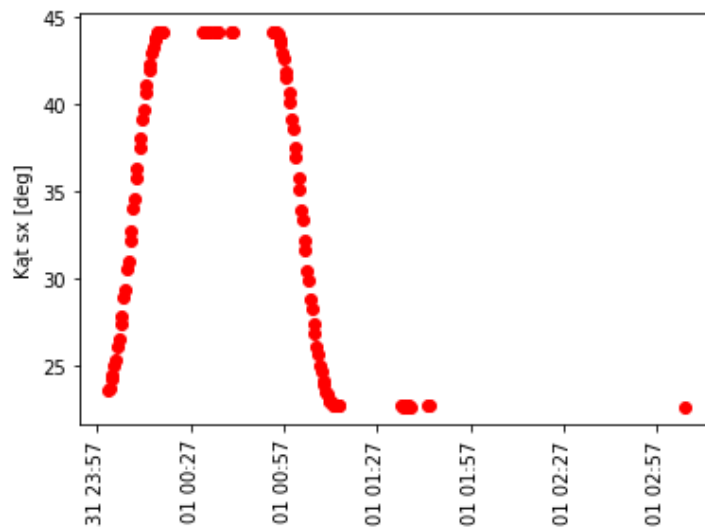
	sa	sx	sy	sz
count	301037	301037	301037	301037
mean	6.28	15.20	89.70	95.30
std	12.38	16.89	8.54	19.49
min	0.00	0.00	1.06	1.63
25%	0.21	4.06	89.80	86.21
50%	1.06	8.64	90.00	93.92
75%	6.02	20.26	90.15	102.66
max	158.38	92.46	179.24	177.97

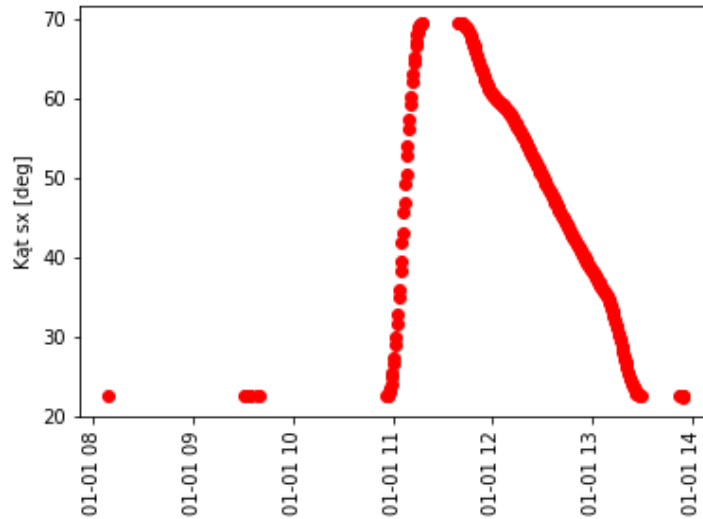
Tabela 2.1: Opis pliku saaf

Rysunek 2.1: Wykres zależności kąta $sa[\text{deg}]$ od czasu

Rysunek 2.2: Wykres zależności kąta $sa[\text{deg}]$ od czasuRysunek 2.3: Wykres zależności kąta $sa[\text{deg}]$ od czasu

Rysunek 2.4 przedstawia wykres zależności kąta sx od czasu. Wykres przypomina nieco rysunek 2.1, widać na nim wyraźną tendencję malejącą do zera i pojedyncze odchylenia. Wykres pierwszych 100 pomiarów kąta sx , przedstawiony na rysunku 2.5 również jest bardzo podobny do rysunku 2.2, gdyż sx wyraża się przez kąt sa . Maksymalna wartość kąta sx między 31 grudnia 2014, godz. 23:57 a 1 stycznia 2015, godz. 3:27, wynosi $44,14^\circ$. Przy tym kącie sonda znajduje się w pozycji optymalnej względem Słońca. Wykres zależności między wartością kąta $sx[\text{deg}]$ a czasem, w godzinach 8:00-14:00 1 stycznia 2015 roku, przedstawiony jest na rysunku 2.6. Podejście do punktu równowagi na rysunku 2.6 jest strome, zaś po godzinie 12:00, gdy wartość kąta sx zaczyna stopniowo opadać, krzywa spadku jest zdecydowanie łagodniejsza.

Rysunek 2.4: Wykres zależności kąta sx [deg] od czasuRysunek 2.5: Wykres zależności kąta sx [deg] od czasu

Rysunek 2.6: Wykres zależności kąta sx [deg] od czasu

Podobne zachowanie przejawiają kąty sy oraz sz , dlatego nie będziemy przedstawiać ich analizy. Jak to było pokazane na przykładzie danych dla sx wszystkie one są powiązane z sa , gdyż panele słoneczne podążały za Słońcem.

2.3.2 Analiza poprawności pomiaru kątów

Plik saaf zawiera informacje odnośnie kąta między panelami słonecznymi a Słońcem (sx , sy , sz) oraz samą sondą i Słońcem (sa). Kąty podane są w stopniach. W tabeli 2.2 przedstawiony jest nagłówek pliku:

Lp.	ut_ms	sa	sx	sy	sz
0	2015-01-01 00:00:32	4.49	23.55	86.77	113.30
1	2015-01-01 00:00:51	3.67	23.74	86.33	113.42
2	2015-01-01 00:01:32	5.71	24.19	85.22	113.66
3	2015-01-01 00:01:51	5.31	24.44	84.69	113.79
4	2015-01-01 00:02:32	7.11	25.05	83.40	114.05

Tabela 2.2: Nagłówek pliku saaf

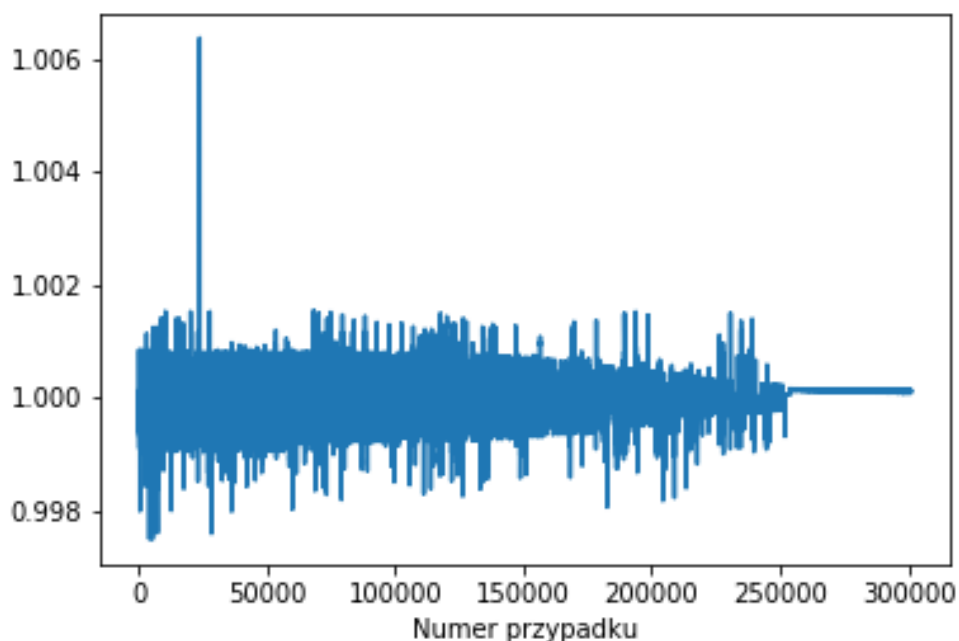
gdzie:

- ut_ms -unix time w milisekundach,
- sa -kąt między normalną do paneli słonecznych sondy a Słońcem,
- sx -kąt między osią X sondy a linią Mars Express-Słońce,
- sy -kąt między osią Y sondy a linią Mars Express-Słońce,
- sz -kąt między osią Z sondy a linią Mars Express-Słońce.

Czas był pierwotnie podany w czasie uniksowym (ang. Unix time), dlatego potrzebna była konwersja dat na bardziej czytelny zapis. Czas uniksowy[42] jest to system zapisu czasu mierzonego w milisekundach od daty 1 stycznia 1970 roku, godz. 00:00:00:00. Jest to tzw. początek epoki Unix.

Jeżeli kąty sx , sy , sz opisują wektor pomiędzy sondą Mars Express a Słońcem, to możemy z nich skonstruować znormalizowany wektor normalny do powierzchni paneli, który spełnia równanie (2.1) - warunek normalizacji wektora.

$$\cos^2(sx) + \cos^2(sy) + \cos^2(sz) = 1 \quad (2.1)$$

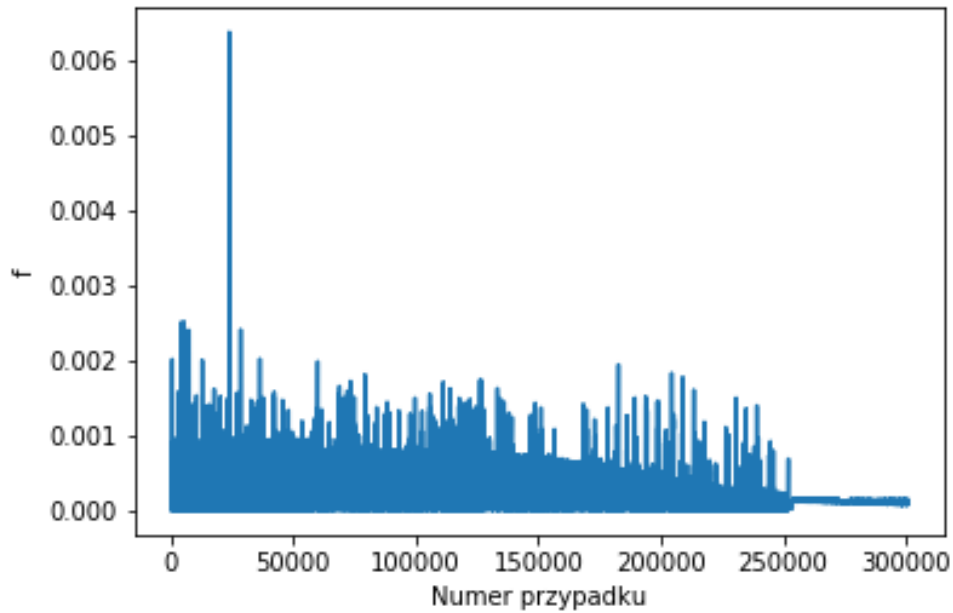


Rysunek 2.7: Wykres sumy kwadratów cosinusów kątów sx , sy , sz

Rysunek 2.7 przedstawia wykres tej zależności w dużej skali. Możemy zauważyć, że wyniki pomiarów kątów są obarczone dużymi niedokładnościami, gdyż norma wspomnianego wektora ma znaczną wariancję wokół wartości 1.

Spróbujemy ocenić maksymalny błąd pomiaru kątów wychylenia paneli przez sondę. Aby łatwiej było to zrobić, przenieśmy wszystkie współczynniki równania na jedną stronę i obłóżmy modulem, tak aby otrzymać:

$$f(sx, sy, sz) = |1 - (\cos^2(sx) + \cos^2(sy) + \cos^2(sz))|. \quad (2.2)$$



Rysunek 2.8: Wykres rozwiązań równania 2.2 dla kątów pliku saaf

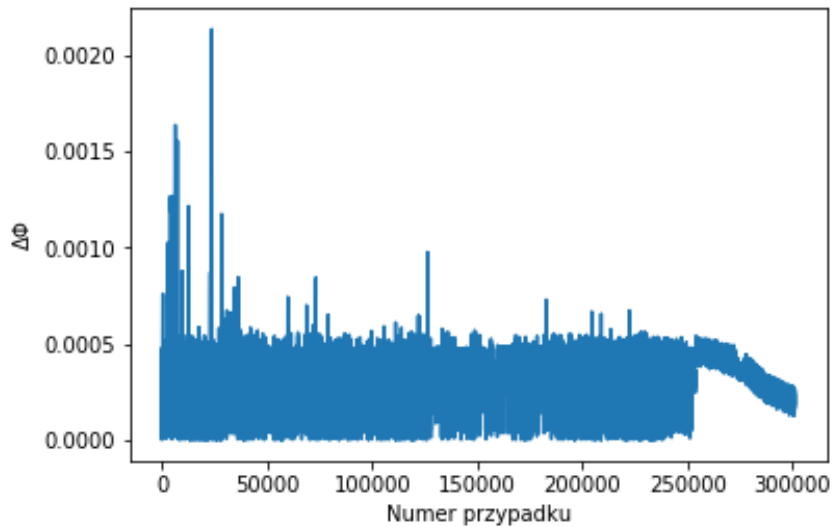
Wzór na propagację błędu przyjmuje dla funkcji z równania 2.2 postać:

$$\Delta f = \sum_{i=1}^3 2 |\cos(\varphi_i) \sin(\varphi_i)| \Delta \varphi_i. \quad (2.3)$$

W celu oszacowania maksymalnego błędu założymy, że każdy z błędów pomiaru kąta jest taki sam $\Delta \varphi_1 = \Delta \varphi_2 = \Delta \varphi_3$. Wówczas obliczamy maksymalny błąd ($\max(\Delta \varphi_i)$) jako:

$$\Delta \varphi = \frac{\Delta f}{\sum_{i=1}^3 2 |\cos(\varphi_i) \sin(\varphi_i)|}, \quad (2.4)$$

gdzie Δf będzie wówczas różnicą pomiędzy wartością na wykresie 2.3.2 a wartością 1.



Rysunek 2.9: Wykres rozwiązań $\Delta\varphi_i$ równania 2.4 dla kątów z pliku saaf

Maksymalna wartość $\Delta\varphi_i$ wynosi 0.00244. Choć nie widać tego wyraźnie na wykresie, w okolicach 0.0025 jest wiele wartości $\Delta\varphi$, więc maksimum o wartości 0.00244 nie można traktować jako pojedynczy artefakt.

Powyższe wyniki pokazują, że danych orientacji paneli słonecznych nie możemy traktować w sposób dokładny. Podano także oszacowanie maksymalnego błędu pomiaru kątów.

2.4 Analiza odległości Słońce-Mars, Ziemia-Mars

Plik ltdata obejmuje dane związane z dynamiką lotu sondy. Opisuje w danym momencie czasu:

- odległość między Słońcem a Marsem
- odległość między Marsem a Ziemią
- kąt Słońce-Mars-Ziemia
- stałą słoneczną dla Marsa [W/m^2]
- czas zaćmienia Słońca przez Mars

W tabelach 2.3 i 2.4 przedstawiony jest nagłówek pliku ltdata.

Lp.	ut_ms	sunmars_km	earthmars_km	sunmarsearthangle_km
0	2015-01-01	2.070435e+08	2.946355e+08	27.68
1	2015-01-02	2.070868e+08	2.953191e+08	27.53
2	2015-01-03	2.071322e+08	2.960026e+08	27.38
3	2015-01-04	2.071797e+08	2.966861e+08	27.23
4	2015-01-05	2.072293e+08	2.973695e+08	27.08

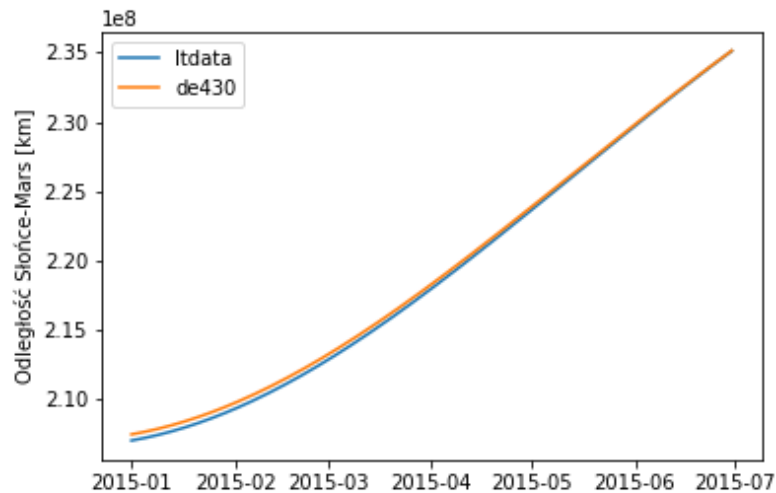
Tabela 2.3: Nagłówek pliku ltdata

Lp.	solarconstantmars	eclipseduration_min	occultationduration_min
0	713.15	37.93	27.17
1	712.85	37.85	27.87
2	712.54	37.72	28.37
3	712.21	37.60	28.95
4	711.87	37.45	29.37

Tabela 2.4: Nagłówek pliku ltdata

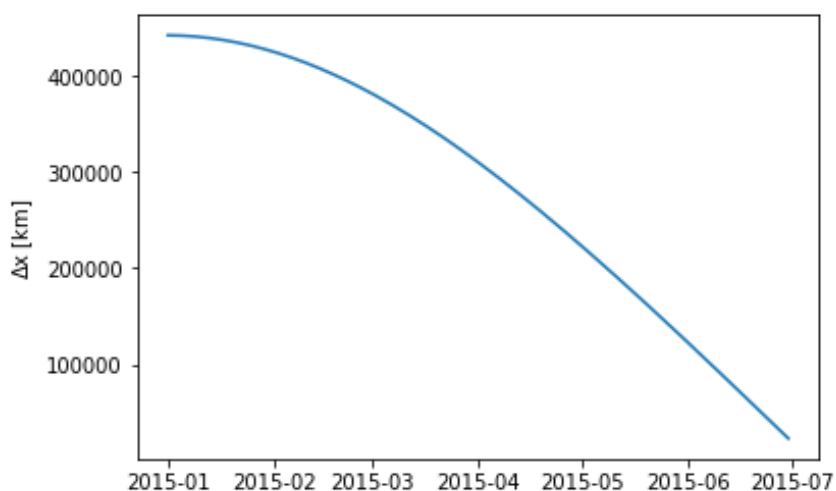
Baza danych de430 pozwala na obliczenie położenia planet względem siebie w dowolnym dniu od roku 1550 do roku 2650. Czas jest obliczany dla niej w według kalendarza juliańskiego, tak więc aby porównać położenia planet zmierzone przez sondę Mars Express i bazę de430, należy najpierw przeprowadzić konwersję czasu. Sonda Mars Express mierzyła czas w Unix Time.

Na początku przeanalizujemy błąd pomiaru odległości Słońce-Mars. Na Rysunku 2.10 przedstawiony jest wykres odległości Słońce-Mars na podstawie danych bazy de430 i Mars Express:



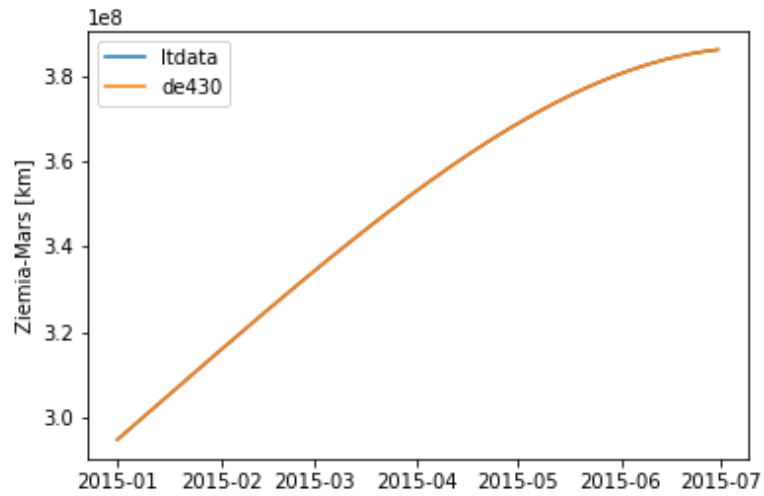
Rysunek 2.10: Odległość Słońce-Mars

Wykres na rysunku 2.10 nie pokazuje dostatecznie dobrze różnic w odległości Słońce-Mars dla obu baz danych. Aby przedstawić ją dokładniej, zamieszczony jest rysunek 2.11 opisujący różnicę Δx w odległości Słońce-Mars dla bazy danych de430 i ltdata. Różnica odległości z czasem maleje, ale nie spada do zera. Jej wartość waha się między 442366 – 22732 km. Biorąc pod uwagę wielkość orbity sondy (zmienną w czasie od 11560km (orbita początkowa) do 10100km (orbita docelowa) [4]) jest to znaczący błąd. Promień Słońca to $6,960 \times 10^8$ km, więc jest to większa odległość niż błąd wyznaczenia pozycji. Dlatego przypuszczalnym źródłem błędów jest niedokładność wyznaczenia centrum Słońca przez systemy sondy. Może się to wiązać z dużym błędem wyznaczania kątów wektora wskazującego pozycję Słońca opisanym w poprzednim rozdziale.

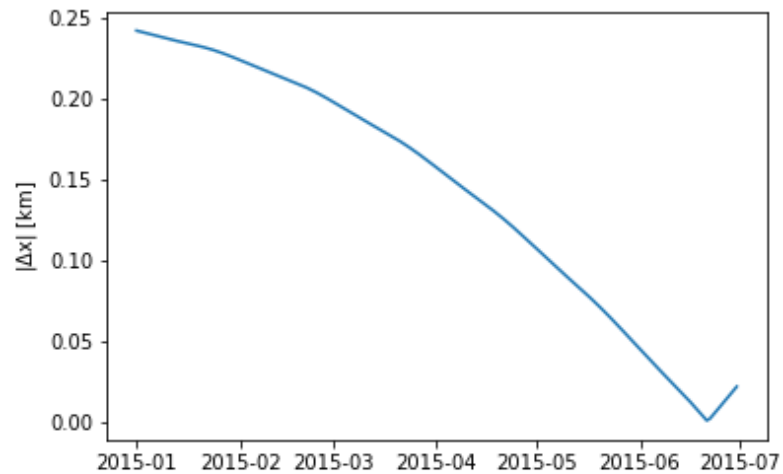


Rysunek 2.11: Różnica odległości Słońce-Mars Δx dla de430 i ltdata

Znacznie lepszą dokładność otrzymuje się dla odległości Ziemia-Mars (Rysunek 2.12). Obie linie tak dobrze się pokrywają, że w efekcie program wykreśla jedynie jedną. Różnica odległości Ziemia-Mars dla obu baz danych jest przedstawiona na Rysunku 2.13.

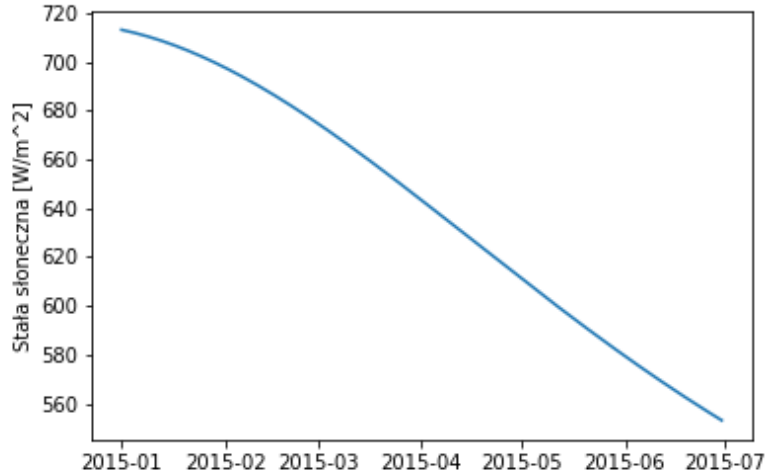


Rysunek 2.12: Odległość Ziemia-Mars

Rysunek 2.13: Różnica odległości Ziemia-Mars Δx dla de430 i ltdata

Różnice w pomiarze odległości Ziemia-Mars wahają się jedynie między 0,25 – 0,00 km, co jeszcze bardziej sugeruje, że pomiar odległości do Słońca był obarczony dużym błędem. Z kolei odległość Ziemia-Mars jest wyznaczona bardzo dokładnie, być może jest to powiązane z faktem, iż położenie Ziemi musiało być znane dokładnie ze względu na komunikację z sondą.

Na kolejnym wykresie (Rysunek 2.14) przedstawiona jest zależność stałej słonecznej od czasu. Jest to ilość energii słonecznej [W/m^2], która dochodzi do Marsa.



Rysunek 2.14: Stała słoneczna [W/m^2]

Wykres zależności stałej słonecznej od czasu przedstawiony jest na rysunku 2.14. Wraz z oddalaniem się Marsa od Słońca, wartość docierającej do niego energii słonecznej maleje. Jest to związane z faktem, że energia jest rozłożona na coraz większych powierzchniach sfer otaczających źródło- Słońce.

2.5 Statystyka wydawania komend

Ostatnim punktem analizy jest próba odpowiedzi na pytania: jaka jest sekwencja wywoływania podsystemów sondy i czy można z analizy tej sekwencji wywnioskować na czym polegały problemy w funkcjonowaniu sondy.

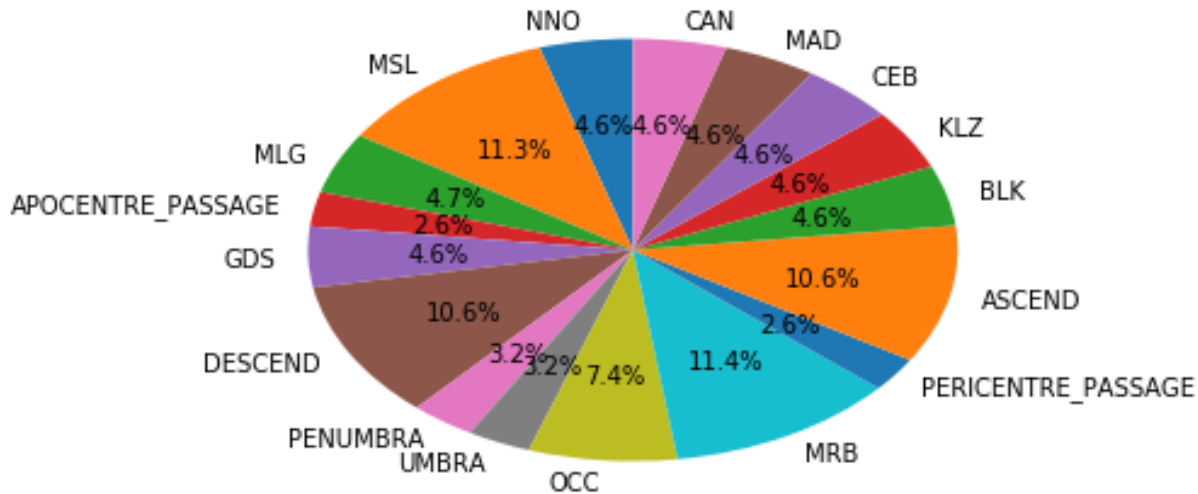
Plik evtf zawiera informacje o wykonywaniu przez podsystemy operacji i czasie ich wykonania. W tabeli 2.5 przedstawiony jest nagłówek pliku.

	ut_ms	description
0	2015-01-01 00:32:06	NNO_AOS_00./_RTLTL_01966
1	2015-01-01 00:39:01	APOCENTRE_PASSAGE_13961
2	2015-01-01 00:55:00	NNO_AOS_05./_RTLTL_01966
3	2015-01-01 01:19:02	NNO_AOS_10./_RTLTL_01966
4	2015-01-01 01:21:47	MSL_AOS_05

Tabela 2.5: Nagłówek pliku evtf

Pełna dokumentacja skrótów wszystkich podsystemów nie jest dostępna publicznie, jednak udało mi się z dostępnej literatury uzyskać informacje na temat kilku. Niektóre z komend są bardzo proste do

objaśnienia, jak np. czas zaćmienia Słońca przez Mars w wyniku orbitowania sondy wokół Marsa. Rysunek 2.15 przedstawia diagram najczęściej wykonywanych komend.



Rysunek 2.15: Diagram częstotliwości padania komend

Najczęściej padające komendy to: MRB, MSL, DESCEND, ASCEND i OCC. Komenda MRB to 11.4% wszystkich komend znajdujących się w pliku evtf. Drugą najczęściej wydawaną komendą jest MSL i stanowi 11.3% komend pliku evtf. Niestety w dokumentacji sondy nie ma informacji o tym jakimi systemami zawiadują te komendy. Kolejnymi komendami, wydawanymi z częstotliwością 10.6% są ASCEND i DESCEND, które odpowiadają za korekcję lotu sondy i odpowiednio wznoszenie i opadanie jej względem powierzchni Marsa. Natomiast komenda OCC wydawana w częstotliwością 7.4% jest skrótem od ang. occultation i wraz z komendami PENUMBRA i UMBRA opisuje zakrycie Słońca przez Mars. Co ciekawe, komendy korygujące lot sondy pojawiały się zdecydowanie częściej niż zasłonięcie Słońca przez Mars. ASCEND i DESCEND to łącznie aż 21,2% wszystkich komend, co pokazuje jak niestabilnym lotem poruszała się sonda w okresie, z którego pochodzą dane. Było to również widoczne w pliku saaf, którego analiza została przedstawiona w poprzednim rozdziale.

2.6 Podsumowanie

W pracy został zaprezentowany sposób gromadzenia danych przez sondę Mars Express. Ze względu na braki w publicznie dostępnej dokumentacji zostały zaprezentowane analizy danych, do których miałam pewność co oznaczają. Zostały wyznaczone niepewności pomiarowe kątów nachylenia paneli słonecznych, następnie przeprowadziłam analizę wykresów ww. kątów. Widać na nich jak bardzo niestabilnym lotem poruszała się sonda. W kolejnym rozdziale porównałam dane dostarczone przez Mars Express dotyczące pomiaru odległości między Marsem a Słońcem i Marsem a Ziemią, z biblioteką astronomiczną NASA de430. Wyniki okazały się dość zgodne, jednak odległość Słońce-Mars ma znacznie większe błędy niż odległość Mars-Ziemia. Ostatnim elementem pracy była statystyka wydawania komend przez podsystemy sondy. W uzyskanych wynikach można zauważyć, że tor lotu sondy był bardzo często korygowany i komendy dotyczące zmiany wysokości lotu zajmują aż 21,2% wszystkich komend.

Z danych wynika, że sonda w okresie, w którym dane zostały zebrane, doświadczała problemów technicznych, szczególnie z efektywnością zbierania energii słonecznej oraz problemy w nawigacji. Celem hackathonu Kaggle była odpowiedź na pytanie, czy coś z takich 'problematycznych' danych możemy wywnioskować. Ta praca jest moją odpowiedzią na tak postawione pytanie.

2.7 Appendix

2.7.1 Kod analizy pliku evtf

Poniżej zamieszczam kod analizy pliku evtf. Kod został wykonany w języku Python, w środowisku Jupyter. W pierwszym kroku analizy pobieram pakiety niezbędne do analizy: pandas, matplotlib oraz datetame.

```
import pandas as pd
import matplotlib.pyplot as plt
iris_filename="Desktop/Fizyka Techniczna/II stopien/mgr/test_set/test_set/
context--2015-01-01_2015-07-01--evtf.csv"
from datetime import datetime
```

Następnie przy użyciu pakietu pandas pobieram plik evtf.csv do obiektu DataFrame, iris. Kolumnę zawierającą czas w Unix Time konwertuję na czas w systemie godzinnym z dokładnością do milisekund. Wyświetlam nagłówek iris.

```
iris=pd.read_csv(iris_filename,sep=",")
iris['ut_ms']= pd.to_datetime(iris['ut_ms'],unit='ms' )
iris.head()
```

Definiuję zmienne, których używam później w pętli zliczającej przypadki powtórzenia się komend w pliku evtf. Po przejściu pętli sprawdzam czy wszystkie komendy zostały zakwalifikowane do jednej z grup.

```
x=0
y=0
z=0
a=0
b=0
c=0
d=0
e=0
f=0
g=0
h=0
i=0
j=0
k=0
m=0
n=0
p=0
for elem in iris['description']:
    if 'NNO' in elem:
        x+=1
    elif 'MSL' in elem:
        y+=1
    elif 'MLG' in elem:
        z+=1
    elif 'APOCENTRE_PASSAGE' in elem:
        a+=1
    elif 'GDS' in elem:
```

```

        b+=1
    elif 'DESCEND' in elem:
        c+=1
    elif 'PENUMBRA' in elem:
        d+=1
    elif 'UMBRA' in elem:
        e+=1
    elif 'OCC' in elem:
        f+=1
    elif 'MRB' in elem:
        g+=1
    elif 'PERICENTRE_PASSAGE' in elem:
        h+=1
    elif 'ASCEND' in elem:
        i+=1
    elif 'BLK' in elem:
        j+=1
    elif 'KLZ' in elem:
        k+=1
    elif 'CEB' in elem:
        m+=1
    elif 'MAD' in elem:
        n+=1
    elif 'CAN' in elem:
        p+=1
    else:
        print(elem)
len(iris['description'])-x-y-z-a-b-c-d-e-f-g-h-i-j-k-m-n-p

```

Tworzę histogram statystyki komend i zapisuję go w pliku hist1.png.

```

plt.figure(figsize=(25,10))
plt.bar(['NNO', 'MSL', 'MLG', 'APOCENTRE_PASSAGE', 'GDS', 'DESCEND', 'PENUMBRA', 'UMBRA', 'OCC', 'MRB',
'PERICENTRE_PASSAGE', 'ASCEND', 'BLK', 'KLZ', 'CEB', 'MAD', 'CAN'], [x,y,z,a,b,c,d,e,f,g,h,i,j,k,m,n,p],
align='center',width=0.5)
plt.savefig('hist1.png')

```

Jako, że histogram nie jest czytelny w stopniu zadowalającym, tworzę diagram kołowy, procentowy statystyki wydawania komend i zapisuję go w pliku evtf.png.

```

labels='NNO', 'MSL', 'MLG', 'APOCENTRE_PASSAGE', 'GDS', 'DESCEND', 'PENUMBRA', 'UMBRA', 'OCC', 'MRB',
'PERICENTRE_PASSAGE', 'ASCEND', 'BLK', 'KLZ', 'CEB', 'MAD', 'CAN'
sizes=[x,y,z,a,b,c,d,e,f,g,h,i,j,k,m,n,p]
plt.pie(sizes, labels=labels,autopct='%1.1f%%',shadow=False, startangle=90)
plt.savefig('evtf.png',bbox_inches='tight')

```

2.7.2 Kod analizy pliku ltdata

Poniżej przedstawiam kod użyty do analizy pliku ltdata. W pierwszym kroku pobieram niezbędne biblioteki: pandas, matplotlib oraz datetime.

```
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
```

Następnie pobieram plik ltdata.csv do obiektu DataFrame, iris. Zmiennej x przypisuję wartość kolumny opisującej czas w pliku ltdata.csv po konwersji czasu z Unix Time na system godzinowy z dokładnością do milisekund. Wyświetlam nagłówek pliku, a następnie tworzę wykresy opisujące dane z pliku, m.in. zależność odległości Ziemia-Mars od czasu i odległości Słońce-Mars od czasu. Wyświetlam dane z ostatnich wierszy pliku.

```
iris_filename="Desktop/Fizyka Techniczna/II stopien/mgr/test_set/test_set/
context--2015-01-01_2015-07-01--ltdata.csv"
iris=pd.read_csv(iris_filename,sep=",")
iris.head()
x=pd.to_datetime(iris['ut_ms'],unit='ms')
y=iris['sunmars_km']
plt.plot(x,y)
y1=iris['earthmars_km']
plt.plot(x,y1)
y2=iris['sunmarsearthangle_deg']
plt.plot(x,y2)
y3=iris['solarconstantmars']
plt.plot(x,y3)
plt.ylabel('Stała słoneczna [W/m^2]')
plt.savefig('solarconstant.png')
y4=iris['eclipseduration_min']
plt.plot(x,y4)
y5=iris['occultationduration_min']
plt.plot(x,y5)
iris.tail()
```

Pobieram pakiet SPK i otwieram plik z biblioteką astronomiczną NASA de430. Importuję pakiet math i przeprowadzam testowe obliczenia. Pakiet i biblioteka działają sprawnie.

```
from jplephem.spk import SPK
kernel = SPK.open('Desktop/Fizyka Techniczna/II stopien/mgr/de430.bsp')
print(kernel)
position = kernel[0,4].compute(2457023.500000)
print(pd.to_datetime(2457023.500000,unit='s'))
print(position)
from math import *
sqrt(position[0]**2+position[1]**2+position[2]**2)
iris['sunmars_km'][0]
```

Tworzę funkcję konwertującą Unix Time na Julian Time oraz funkcję obliczającą długość wektora odległości. Tworzę pustą listę odległości Słońca od Marsa. W pętli po elementach kolumny opisującej czas

w DataFrame `iris`, wywołuję funkcję konwertującą Unix Time, w którym zapisany jest czas w `iris`, na Julian Time, którym posługuje się biblioteka astronomiczna `de430`. Następnie wywołuję element biblioteki opisujący wzajemne położenie Słońca i Marsa dla zadanej daty. Jako że biblioteka astronomiczna `de430` przechowuje wektory odległości, do celów naszej analizy konieczne jest wywołanie funkcji obliczającej długość wektora odległości Mars-Słońce. Wartość tej długości jest zapisywana w kolejnym elemencie listy `sunmars`. Po przejściu pętli wyświetlam wykres porównujący odległość Mars-Słońce dla pliku `ltdata` i biblioteki `de430`. Zapisuję wykres w pliku `sunmars.png`.

```
def getJulianFromUnix(unixSecs):
    return ( unixSecs / 86400000 ) + 2440587.5
def distance(a,b,c):
    return sqrt(a**2+b**2+c**2)
sunmars=[]
for item in iris['ut_ms']:
    date=getJulianFromUnix(item)
    position= kernel[0,4].compute(date)
    sunmars.append(distance(position[0],position[1],position[2]))
plt.plot(x,y,x,sunmars)
plt.legend(['ltdata','de430'])
plt.ylabel('Odległość Słońce-Mars [km]')
plt.savefig('sunmars.png')
```

Podobną procedurę powtarzam dla odległości Mars-Ziemia.

```
earthmars=[]
for item in iris['ut_ms']:
    date=getJulianFromUnix(item)
    position = kernel[0,4].compute(date)
    position -= kernel[0,3].compute(date)
    position -= kernel[3,399].compute(date)
    earthmars.append(distance(position[0],position[1],position[2]))
plt.plot(x,y1,x,earthmars)
plt.legend(['ltdata','de430'])
plt.ylabel('Ziemia-Mars [km]')
plt.savefig('earthmars.png')
```

W ostatnim kroku porównuję odległości Ziemia-Mars i Słońce-Mars dla pliku `ltdata` i biblioteki `de430` poprzez obliczenie różnicy odległości obu wartości, wyświetlenie ich na wykresie i zapisanie do pliku.

```
roznica1=[]
for i in range(len(y)):
    roznica1.append(abs(y[i]-sunmars[i]))
plt.plot(x,roznica1)
plt.ylabel(r'$\Delta x$ [km]')
plt.savefig('roznica1.png',bbox_inches='tight')
min(roznica1)
roznica2=[]
for i in range(len(y1)):
    roznica2.append(abs(y1[i]-earthmars[i]))
plt.plot(x,roznica2)
plt.ylabel(r'$|\Delta x|$ [km]')
plt.savefig('roznica2.png')
```

2.7.3 Kod analizy pliku saaf

Poniżej zamieszczam kod analizy pliku saaf. W pierwszym kroku pobieram potrzebne pakiety: pandas, matplotlib, datetime, math, time z astropy i elementy pakietu poliastro. Pobieram plik saaf.csv do obiektu DataFrame, sprawdzam podstawowe właściwości saaf przez komendy head(), tail(), describe(), columns. Konwertuję czas z odpowiedniej kolumny z Unix Time na system godzinowy z dokładnością do milisekund.

```
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from math import *
import astropy.units as u
from astropy import time

from poliastro import iod
from poliastro.bodies import Sun
from poliastro.twobody import Orbit

from poliastro.util import time_range
iris_filename="Desktop/Fizyka Techniczna/II stopien/mgr/test_set/test_set/
context--2015-01-01_2015-07-01--saaf.csv"
iris=pd.read_csv(iris_filename,sep=",")
iris['ut_ms']= pd.to_datetime(iris['ut_ms'],unit='ms' )
iris.head()
iris.tail()
iris.describe()
iris.columns
x=iris['ut_ms']
y=iris['sa']
df=pd.DataFrame(iris['ut_ms'])
df['ut_ms'] = pd.to_datetime(df['ut_ms'],unit='ms')
df.head()
```

Wyświetlam i zapisuję wykresy dla różnych zakresów kąta *sa* w zależności od czasu. Podobne operacje wykonuję dla kątów *sx*, *sy*, *sz* z pliku saaf. Wyszukuję maksima i minima na wykresach.

```
plt.plot(df['ut_ms'],y)
plt.ylabel('Kąt sa [deg]')
plt.savefig('sa1.png')
max(y[0:100])
plt.plot(df['ut_ms'][0:100],y[0:100], 'ro')
plt.ylabel('Kąt sa [deg]')#[rad]')
#plt.xlabel('Pierwsze 100 przypadkow kata sa')
plt.xticks(rotation=90)
plt.savefig('sa3.png',bbox_inches='tight')
plt.plot(df['ut_ms'][450:600],y[450:600], 'ro')
plt.ylabel('Kąt sa [deg]')#[rad]')
plt.xticks(rotation=90)
plt.savefig('sa450600.png',bbox_inches='tight')
len(df['ut_ms'])
```



```

y1=iris['sx']
plt.plot(df['ut_ms'],y1)#*(3.14/180.)
plt.ylabel('Kąt sx [deg]')
plt.savefig('sx.png')
plt.plot(df['ut_ms'][0:100],y1[0:100],'ro')
plt.ylabel('Kąt sx [deg]')
plt.xticks(rotation=90)
plt.savefig('sx1.png',bbox_inches='tight')
max(y1[0:100])
plt.plot(df['ut_ms'][450:730],y1[450:730],'ro')
plt.ylabel('Kąt sx [deg]')
plt.xticks(rotation=90)
plt.savefig('sx2.png',bbox_inches='tight')
y2=iris['sy']
plt.plot(df['ut_ms'],y2)
plt.ylabel('Kąt sy [deg]')
plt.savefig('sy.png')
plt.plot(df['ut_ms'][0:100],y2[0:100],'ro')
plt.ylabel('Kąt sy [deg]')
plt.xticks(rotation=90)
plt.savefig('sy1.png',bbox_inches='tight')
plt.plot(df['ut_ms'][450:730],y2[450:730],'ro')
plt.ylabel('Kąt sy [deg]')
plt.xticks(rotation=90)
plt.savefig('sy2.png',bbox_inches='tight')
y3=iris['sz']
plt.plot(df['ut_ms'],y3)
plt.ylabel('Kąt sz [deg]')
plt.savefig('sz.png')
pr=len(y3)-1000
min(y3[pr:])
plt.plot(df['ut_ms'][pr:],y3[pr:])
plt.plot(df['ut_ms'][0:100],y3[0:100],'ro')
plt.ylabel('Kąt sz [deg]')
plt.xticks(rotation=90)
plt.savefig('sz1.png',bbox_inches='tight')
max(y3[0:100])
plt.plot(df['ut_ms'][450:730],y3[450:730],'ro')
plt.ylabel('Kąt sz [deg]')
plt.xticks(rotation=90)
plt.savefig('sz2.png',bbox_inches='tight')

```

Rozpoczynam analizę częstotliwości występowania picków w wykresach *sa*, która ma na celu obliczenie czasu okrążenia Marsa przez sondę. Wyświetlam wykresy 100, 600 i 1200 wartości *sa*.

```

plt.plot(df['ut_ms'][0:100],y[0:100],'ro')
plt.ylabel('Kąt sa [deg]')#[rad]')
#plt.xlabel('Pierwsze 100 przypadkow kata sa')
plt.xticks(rotation=90)
plt.savefig('sa3.png',bbox_inches='tight')

```

```
plt.plot(df['ut_ms'][0:600],y[0:600],'ro')
plt.xticks(rotation=90)
plt.ylabel('Kąt sa [deg]')
plt.savefig('25_26.png',bbox_inches='tight')
```

```
plt.plot(df['ut_ms'][0:1200],y[0:1200],'ro')
plt.xticks(rotation=90)
plt.ylabel('Kąt sa [deg]')
plt.savefig('1200_przypadkowsa.png',bbox_inches='tight')
```

Wybieram wykres 1200 kątów *sa* jako bazę do analizy. Tworzę puste listy, do których zapisuję wyniki działania pętli wyszukującej picków. Wyświetlam wykres czasu występowania picków. Do dalszych operacji konieczne jest pobranie pakietu NumPy. Na podstawie wykresu czasu występowania picków, dzielę wykres 1200 kątów *sa* na pięć części, w których wyszukuję maksima lokalne picków oraz czas ich występowania.

```
listadat=[]
indeks=[]
listakatow=[]
listalaczna=[]
for i in range(1200):
    if y[i] >= 20:
        listadat.append(df['ut_ms'][i])
        listakatow.append(y[i])
        indeks.append(i)
        listalaczna.append([df['ut_ms'][i],y[i]])
plt.plot(listadat,'ro')
max(listakatow[0:50])
import numpy as np
listadatpick=[]
listapick=[]
listadatpick.append(listadat[np.argmax(listakatow[0:50])])
listadat[np.argmax(listakatow[0:50])]
listapick.append(max(listakatow[0:50]))
print(max(listakatow[50:170]))
print(listadat[50+np.argmax(listakatow[50:170])])
listadatpick.append(listadat[50+np.argmax(listakatow[50:170])])
listapick.append(max(listakatow[50:170]))
print(max(listakatow[170:220]))
print(listadat[170+np.argmax(listakatow[170:220])])
listadatpick.append(listadat[170+np.argmax(listakatow[170:220])])
listapick.append(max(listakatow[170:220]))
print(max(listakatow[220:280]))
print(listadat[220+np.argmax(listakatow[220:280])])
listadatpick.append(listadat[220+np.argmax(listakatow[220:280])])
listapick.append(max(listakatow[220:280]))
print(max(listakatow[280:]))
print(listadat[280+np.argmax(listakatow[280:])])
listadatpick.append(listadat[280+np.argmax(listakatow[280:])])
listapick.append(max(listakatow[280:]))
```

```
print(listadatpick)
print(listapick)
```

Obliczam niepewność pomiarową kątów pliku saaf. Wyniki wyświetlam na wykresach i zapisuję. Wyszukuję maksymalne odchylenie standardowe.

```
err2=[]
for i in range(len(y1)):
    err2.append(cos(y1[i]*3.14/180.)**2+cos(y2[i]*3.14/180.)**2+cos(y3[i]*3.14/180.)**2)
plt.plot(err2)
plt.xlabel('Numer przypadku')
plt.savefig('err2.png')
err3=[]
for i in range(len(y1)):
    err3.append(abs(1-err2[i]))
plt.plot(err3)
plt.xlabel('Numer przypadku')
plt.ylabel('f')
plt.savefig('err3.png',bbox_inches='tight')
delta_fi=[]
for i in range(len(err3)):
    delta_fi.append(err3[i]/(2*abs(cos(y1[i]*3.14/180)
    *sin(y1[i]*3.14/180))+2*abs(cos(y2[i]*3.14/180)
    *sin(y2[i]*3.14/180))+2*abs(cos(y3[i]*3.14/180)
    *sin(y3[i]*3.14/180))))
plt.plot(delta_fi)
plt.xlabel('Numer przypadku')
plt.ylabel('$\Delta \Phi$')

plt.savefig('err4.png',bbox_inches='tight')
max(delta_fi)
delta_fi.remove(0.0024355056738007393)
max(delta_fi)
delta_fi.remove(0.002290192846866648)
max(delta_fi)
delta_fi.remove(0.002215832652615082)
max(delta_fi)
```


Bibliografia

- [1] <http://sci.esa.int/mars-express/> (data dostępu: 08.09.2018)
- [2] National Geographic, *Space Probes*, <https://www.nationalgeographic.org/media/space-probes/> (data dostępu: 08.09.2018)
- [3] Planetary, *Planetary Exploration Timelines: A Look Ahead to 2016* <http://www.planetary.org/blogs/emily-lakdawalla/2015/12311322-planetary-exploration-timelines.html> (data dostępu: 08.09.2018)
- [4] R. Pischel, T. Zegers, *Mars Express Science Planning and Operations*, ESA SP-1291, (2009)
- [5] S. Thornton, J. Marion, *Classical Dynamics of Particles and Systems*, Brooks Cole, 2003, ISBN 0-534-40896-6
- [6] https://commons.wikimedia.org/wiki/File:Hohmann_transfer_orbit.png (data dostępu: 08.09.2018)
- [7] Basics of Space Flight. NASA, *Section 1: Environment, Chapter 4: Trajectories*, <https://solarsystem.nasa.gov/basics/chapter4-1/> (data dostępu: 08.09.2018)
- [8] S. D. Ross, *The Interplanetary Transport Network*, American Scientist. 94: 230-237, 2006
- [9] Misje Sond Kosmicznych, *Sondy Planetarne*, <https://www.misjesondkosmicznych.pl/sondyplanetarne.html> (data dostępu: 08.09.2018)
- [10] NASA, *Europa Clipper*, <https://www.nasa.gov/europa> (data dostępu: 08.09.2018)
- [11] <https://www.nasa.gov/europa/overview/index.html> (data dostępu: 08.09.2018)
- [12] <https://encyklopedia.pwn.pl/haslo/Mars;3938054.html> (data dostępu: 08.09.2018)
- [13] https://www.esa.int/spaceinimages/Images/2018/07/Stormy_Mars_in_opposition_in_2018 (data dostępu: 08.09.2018)
- [14] <https://mars.nasa.gov/allaboutmars/mystique/history/early/> (data dostępu: 08.09.2018)
- [15] <https://mars.nasa.gov/allaboutmars/mystique/history/1500/> (data dostępu: 08.09.2018)
- [16] <https://mars.nasa.gov/allaboutmars/mystique/history/1600/> (data dostępu: 08.09.2018)
- [17] <https://mars.nasa.gov/allaboutmars/mystique/history/1700/> (data dostępu: 08.09.2018)
- [18] <https://mars.nasa.gov/allaboutmars/mystique/history/1800/> (data dostępu: 08.09.2018)

- [19] <https://mars.nasa.gov/programmissions/missions/> (data dostępu: 08.09.2018)
- [20] http://www.esa.int/spaceinimages/Images/2014/10/Mars_Express (data dostępu: 08.09.2018)
- [21] https://www.esa.int/Our_Activities/Space_Science/Mars_Express_overview (data dostępu: 08.09.2018)
- [22] http://www.esa.int/Our_Activities/Space_Science/Mars_Express/Beagle-2_lander_found_on_Mars (data dostępu: 08.09.2018)
- [23] http://www.esa.int/Our_Activities/Space_Science/Mars_Express/Phobos_flyby_images (data dostępu: 08.09.2018)
- [24] http://www.esa.int/spaceinimages/Images/2010/03/Mars_Express_HRSC_image_of_Phobos_taken_on_7_March_2010 (data dostępu: 08.09.2018)
- [25] Burns, A. Joseph, *Contradictory Clues as to the Origin of the Martian Moons*, University of Arizona Press, Tucson, 1992
- [26] Geoffrey A. Landis, *Origin of Martian Moons from Binary Asteroid Dissociation*, Paper AAAS - 57725, American Association for Advancement of Science Annual Meeting, Boston, 2002
- [27] A. Boschetti, L. Massaron, *Python. Podstawy nauki o danych*, wyd. II, Helion, 2017
- [28] <http://www.numpy.org/> (data dostępu: 08.09.2018)
- [29] <https://www.scipy.org/> (data dostępu: 08.09.2018)
- [30] <https://pandas.pydata.org/> (data dostępu: 08.09.2018)
- [31] <https://matplotlib.org/> (data dostępu: 08.09.2018)
- [32] <http://jupyter.org/> (data dostępu: 08.09.2018)
- [33] <https://pypi.org/project/jplephem/> (data dostępu: 08.09.2018)
- [34] <https://naif.jpl.nasa.gov/naif/> (data dostępu: 08.09.2018)
- [35] https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/ (data dostępu: 08.09.2018)
- [36] https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/ (data dostępu: 08.09.2018)
- [37] https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/aareadme_de430-de431.txt (data dostępu: 08.09.2018)
- [38] https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/de430_and_de431.pdf (data dostępu: 08.09.2018)
- [39] <https://www.kaggle.com/kaggle> (data dostępu: 08.09.2018)
- [40] <https://www.kaggle.com/learn/overview> (data dostępu: 08.09.2018)
- [41] <https://www.kaggle.com/c/mars-express-power-hackathon> (data dostępu: 08.09.2018)
- [42] <https://www.epochconverter.com/> (data dostępu: 08.09.2018)