The Challenges of Sending Astronauts to Mars

Jered Dominguez-Trujillo

“Elon is changing the way aerospace business is done” (p. 256, Vance)

INTRODUCTION

At the age of 14, Elon Musk had an existential crisis that would enlighten him on his purpose in life, which he has said is to “be well on our way to becoming a multiplanetary species with a self-sustaining civilization on another planet… I think that would be really good” (p. 5, Vance). His founding of SpaceX in 2002 served the beginning of the fulfillment of this purpose.

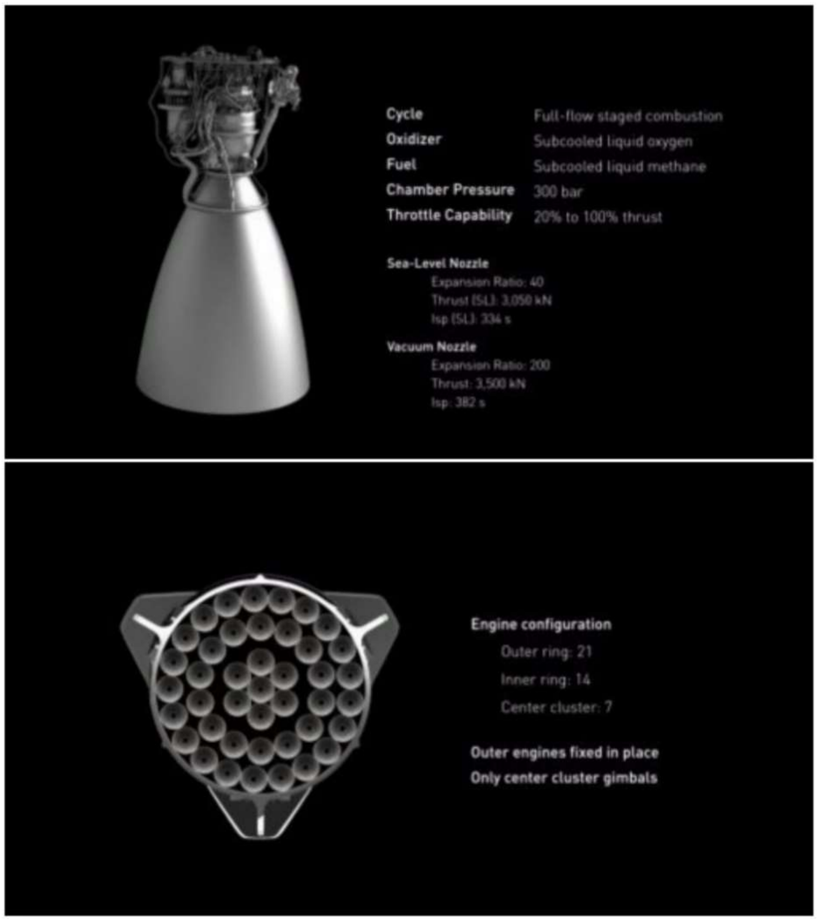
Since SpaceX was founded in 2002, it has developed several rocket engines, rocket boosters, and spaceships that have enabled it to be the first private company to launch its own rockets into Low-Earth Orbit (LEO) and to send supplies to the International Space Station (ISS). The development of the Merlin engines, Falcon 1 and Falcon 9 rockets, and the Dragon capsule landed the company a lucrative contract with NASA to send cargo to the ISS over the course of several years. This was possible with the large amount of capital ($100 million) Elon Musk invested in the company to get it off the ground, and despite nearly going bankrupt amid several launch delays and failures, SpaceX successfully launched its first Falcon 1 rocket on September 28, 2008, after three previous failed attempts. Since then, the company has developed the Falcon 9 and Falcon Heavy rockets, and it was revealed in recent years that work on the new Raptor engine, Dragon capsule and the Interplanetary Transport System were underway with the goal of sending astronauts to the ISS and eventually to Mars and beyond.

Since the initial launch of the Falcon 1 rocket in 2008, SpaceX had successfully flown the more powerful Falcon 9 rocket with a simulated Dragon payload in June 2010, and successfully delivered cargo to the ISS in May 2012. Since then, SpaceX has regularly flown cargo missions to the ISS, but not without failure. A catastrophic explosion of a Falcon 9 rocket in 2015 and occasional leaks and delays have been issues for the company, preventing the desired delivery rate to the ISS from becoming a reality. These difficulties have awakened Elon Musk to the challenges of spaceflight, but have not dissuaded him from the goal of reaching Mars and becoming an “Interplanetary species.”

In 2011, Elon Musk expressed to scientists at the American Institute of Aeronautics and Astronautics (AIAA) that he envisioned sending astronauts to Mars in the next 15 years. This led to the development of reusable rocket technology, which SpaceX has successfully tested on water and land. Further developments appeared to be taking place within the company to continue to develop aerospace technologies that would enable the flight of a crewed mission to Mars.

The new Raptor engines developed by SpaceX and recently tested at the Stennis Space Center with the objective of developing a rocket engine that would have enough propulsive power to send a fully manned and supplied crew from Earth to Mars. The engine is the first full flow, Methane-Liquid Oxygen, staged combustion engine tested, and the first full-flow design tested since the Russian RD-270. A unique test stand that could generate extremely high-pressure liquid and hot gas of the fuel and oxidizer simultaneously was required to test the full flow engine. The engine is fueled by sub-cooled liquid oxygen and methane, consists of a multi-state turbopump and has a record chamber pressure of $300\ bar$, much higher than the maximum pressure of $267\ bar$ achieved by the RD-180 engine used on the Atlas V and higher than the $216\ bar$ chamber pressure of the Space Shuttle Main Engine (SSME) nozzle, and is projected to have two versions.

The Raptor Sea-Level version is expected to have a thrust of $3{,}050\ kN$ and specific impulse of $334\ seconds$ at sea level and a thrust of $3{,}297\ kN$ and specific impulse of $361\ seconds$ at vacuum with a nozzle area ratio of 40, resulting in an approximately $1.7\ m$ nozzle exit diameter. The Raptor Vac version is expected to have a much larger nozzle area ratio of 200 (resulting in a $\sim 4\ m$ nozzle exit diameter), resulting in a thrust of $3{,}500\ kN$ and specific impulse of $382\ seconds$ at vacuum conditions. Each version is expected to be identical other than the extended nozzle on the Raptor Vac that optimizes thrust in vacuum conditions, to streamline the production process. The Interplanetary



Cycle	Full-flow staged combustion
Oxidizer	Subcooled liquid oxygen
Fuel	Subcooled liquid methane
Chamber Pressure	300 bar
Throttle Capability	20% to 100% thrust

Sea-Level Nozzle
- Expansion Ratio: 40
- Thrust (SL): 3,050 kN
- Isp (SL): 334 s

Vacuum Nozzle
- Expansion Ratio: 200
- Thrust: 3,500 kN
- Isp: 382 s

Engine configuration
- Outer ring: 21
- Inner ring: 14
- Center cluster: 7

Outer engines fixed in place
Only center cluster gimbals

Figure 1: Top: CAD model and specifications of Raptor Engine.
Bottom: Raptor engines configured to fit within the cavity of the ITS Booster.
Credit: SpaceX
Top: http://spacelfight101.com/spx/wp-content/uploads/sites/113/2016/09/ITS-019-1024x576.jpg
Bottom: http://spaceflight101.com/spx/wp-content/uploads/sites/113/2016/09/ITS-021-1024x576.jpg

Transport System (ITS) that Elon Musk has envisioned utilizing the Raptor engines to carry passengers from Earth to Mars is projected to be outfitted with a first stage containing 42 Raptor engines, producing a total upwards thrust of $\sim 130\ MN$. A second stage is expected to have 9 Raptor engines, and further fueling, reusability, and return complications have been outlined but specific details have not been provided at the time of writing. There have been mixed reports about thrust vectoring capabilities and other gimbal functions, and it is currently expected that the central 9 Raptor engines will be gimbaled to enable thrust vectoring. Testing and successfully firing many such highly integrated full-flow engines that operate at extremely high chamber pressures in sync and without any malfunctions, or leaks presents an unprecedented challenge for the company.

Musk projects that eventually trips to Mars will take less than 80 days, and may become as fast as 30 days as propulsion technologies advance, a large reduction from the popular and fuel efficient ~ 8 month travel time available from a near Hohmann transfer from Earth to Mars today. He also is planning an initial unmanned mission to Mars in 2018, and is hopeful that the company will be able to send a manned crew to Mars by 2024. SpaceX has yet to perform a manned mission or perform a mission that is outside of LEO in the 15 years it has been functioning, this timeline is rather aggressive, characteristic of Elon Musk's ambition and optimism. The following sections are to discuss the technical aspects of performing a near Hohmann transfer orbit from Earth to Mars in 3-dimensions, and what steps would be necessary to reduce the travel time between Earth and Mars from $\sim 8\ months$, to less than $80\ days$. The Raptor engine performance will be analyzed and compared to rough MATLAB numerical calculations and engine performance (thrust coefficient) at various exit pressures will be graphically presented. Further technical details and financial challenges that the company should expect to face are briefly discussed. The following sections will analyze a subset of the challenges and feasibility of the missions outline by Elon Musk and SpaceX as the company prepares to begin manned missions and eventually a mission to Mars.

ANALYSIS

Raptor Engine Nozzle

The Raptor engine has been specified to have two separate versions with different nozzle exit area ratios of 40 and 200, for the Sea-Level optimized version and the Vacuum optimized version, respectively. Both versions have been confirmed to operate at a record chamber pressure of $300\ bar$, and are full-flow engines that are fueled by sub-cooled methane and oxygen. They must provide enough thrust to transport $450\ tons$ of payload from Earth to Mars, must be reusable, and must be able to refuel both in orbit and on Mars. These specifications outline the most sophisticated rocket engine ever built, and Elon Musk plans on building 51 per ITS spacecraft. The engines must function for long periods of time, under extreme conditions, and simultaneously for the planned mission to be a success and to safely deliver humans to Mars.

A MATLAB script was written to analyze the performance of the nozzle given the area ratio, an assumed inlet temperature of $3500\ K$, and an exit diameter, specified as $\sim 1.7\ m$ for the sea-level version and $\sim 4\ m$ for the vacuum optimized version of the engine. The design pressure and altitude were then numerically calculated. The thrust at sea-level, at design altitude, and at vacuum conditions was then calculated. The resulting thrusts are then compared to the official Raptor specifications and discussed. The governing one-dimensional gas dynamic equations are outlined below:

$$\frac{A_e}{A^*} = [\frac{2 + (\gamma - 1)M_e^2}{\gamma + 1}]^{\frac{\gamma+1}{2(\gamma+1)}} \qquad \textbf{\textit{Eq. 1}}$$

$$\frac{T_o}{T_e} = \left(\frac{P_o}{P_e}\right)^{\frac{\gamma-1}{\gamma}} \qquad \textbf{\textit{Eq. 2}}$$

$$P_e\left(1 + \frac{\gamma - 1}{2}M_e^2\right)^{\frac{\gamma}{\gamma-1}} = P_o \qquad \textbf{\textit{Eq. 3}}$$

$$Thrust\ Coefficient = C_f = \sqrt{\frac{2\gamma}{\gamma - 1}(\frac{2}{\gamma + 1})^{\frac{\gamma+1}{\gamma-1}}[1 - (\frac{P_e}{P_o})^{\frac{\gamma-1}{\gamma}}]} + \left(\frac{P_e}{P_o} - \frac{P_a}{P_o}\right) * \frac{A_e}{A^*} \qquad \textbf{\textit{Eq. 4}}$$

$$C_f = \frac{Thrust}{P_o * A^*} \qquad \textbf{\textit{Eq. 5}}$$

$$Thrust \text{ @ Design Pressure} = T_d = C_{f,max} * P_o * A^* \qquad \textbf{\textit{Eq. 6}}$$

$$Thrust = T_d - (P_a - P_e)A_e \qquad \textbf{\textit{Eq. 7}}$$

$$Altitude = h = T_{ref}\frac{\left[1 - \left(\frac{P_e}{P_{ref}}\right)^{\frac{1}{5.2561}}\right]}{0.0065} \qquad \textbf{\textit{Eq. 8}}$$

Where $T_{ref} = 288.15\ K$ and $P_{ref} = 101325\ Pa$

The method employed within the MATLAB program is described as follows. An inlet temperature of $3500\ K$ was assumed and the nozzle area ratios were given for each version of the Raptor engine. For an exit diameter of $1.7\ m$ and an area ratio of 40, the nozzle area was calculated to be $0.0629\ m^2$. This nozzle area was then used to calculate the exit area of both versions of the engine, utilizing the area ratio. The design pressure was incremented by $10\ Pa$ and the thrust coefficient curve was calculated. This was repeated until the maximum of the thrust coefficient curve matched the specified area ratio of the nozzle, defining the design pressure. The design altitude, design thrust, sea-level thrust, and vacuum thrust were then calculated. The resulting thrust coefficient design curves of each version of the engine are presented below.

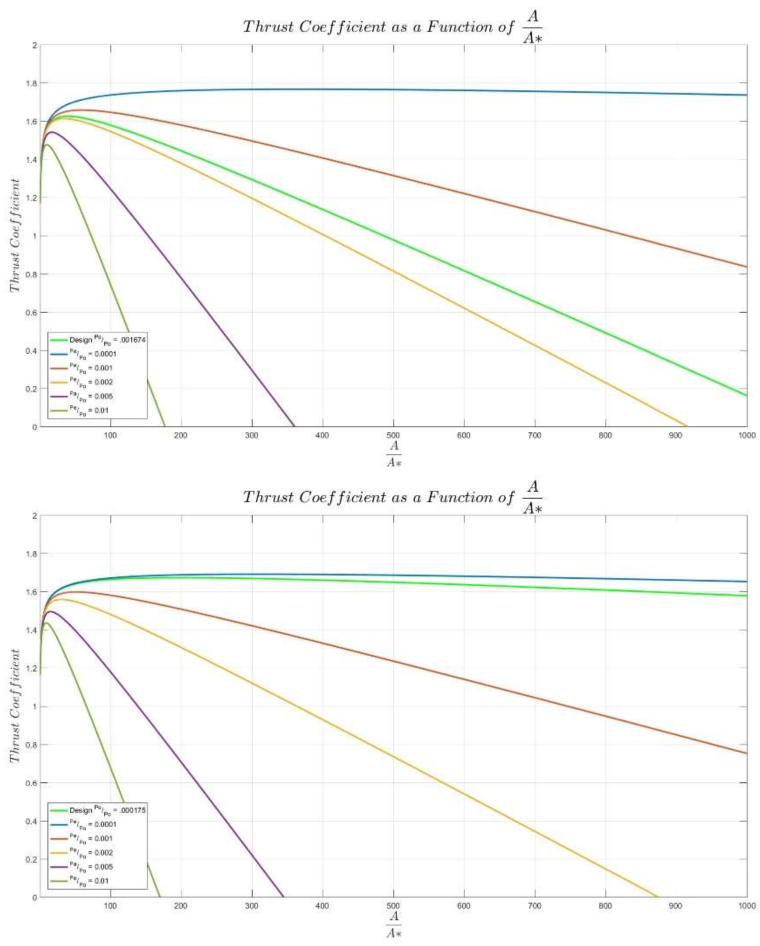


Figure 2: Top: Raptor Engine Thrust Coefficient at an area ratio of 40. Bottom: Raptor Engine Thrust Coefficient at an area ratio of 200. Generated by MATLAB.

Figure 2 indicated the variation in thrust coefficient based on the two versions of the proposed Raptor engine. It was observed that the nozzle with the higher area ratio resulted in a lower design pressure (higher design altitude), as would be expected. The higher area ratio also provided a larger maximum thrust coefficient (~ 1.8) compared to the maximum thrust coefficient of the low area ratio nozzle (~ 1.6) that was designed for lower altitudes. This resulted in the sea-level design capable of providing a thrust of $2{,}949\ kN$ at sea level compared to the $1{,}959\ kN$ provided by the vacuum design. However, the vacuum design outperformed the sea-level optimization at the design altitude with a calculated design thrust of $3{,}168\ kN$ at an altitude of $19.09\ km$, compared to the calculated design thrust of $3{,}077\ kN$ at an altitude of $5.52\ km$. The specified and calculated data are presented and compared below. The specified data were obtained from http://spaceflight101.com/spx/spacex-raptor/

Table 1: Raptor Engine Performance

	Raptor Sea-Level		Raptor Vacuum	
	Specified	**Calculated**	**Specified**	**Calculated**
Chamber Pressure	$300\ bar$			
Area Ratio	40		200	
Thrust (Sea Level)	$3{,}050\ kN$	$2{,}949\ kN$		$1{,}959\ kN$
Thrust (Vacuum)	$3{,}297\ kN$	$3{,}204\ kN$	$3{,}500\ kN$	$3{,}233\ kN$
Specific Impulse (Sea Level)	$334\ seconds$			
Specific Impulse (Vacuum)	$361\ seconds$		$382\ seconds$	
Design Pressure		$50380\ Pa$		$5250\ Pa$
Design Altitude (Earth)		$5.52\ km$		$19.09\ km$
Design Thrust		$3{,}077\ kN$		$3{,}168\ kN$

The variation in design altitude, pressure and thrust as a function of area ratio is apparent. The calculated figures indicated some variance from the provided specifications, which is largely accounted for by approximations made in the applied MATLAB code to perform the necessary calculations with a reasonable amount of computer resources. The large chamber pressure of $300\ bar$ proposed for the Raptor engines enables it to produce extraordinary amounts of thrust with a nozzle similar in size to the Merlin engine nozzle and the SSME engine nozzle. The challenges in accurately calculating and measuring engine performance are inherent to any propulsive engine calculation and SpaceX is more than capable of producing a capable engine. The challenge will lie in the testing of the full-flow Raptor engines at the record high chamber pressure, in the mass production of the engines to outfit each ITS with 42 engines on the first stage and 9 engines on the second stage, in firing all engines simultaneously during launch, in ensuring functionality, full maintenance and no leakage in any of the hardware, and in funding the massive project of producing a sufficient amount of engines to provide for the construction of multiple ITS vehicles. The ambitious timeline of producing a reusable, affordable, and mass produced version of the Raptor engine to enable human travel to and return from Mars by the year 2024 requires a substantial financial and man power commitment that Elon Musk is striving for, but will likely be late upon delivery, as has been customary in his projects across all his companies.

Orbital Trajectory – Near Hohmann Transfer

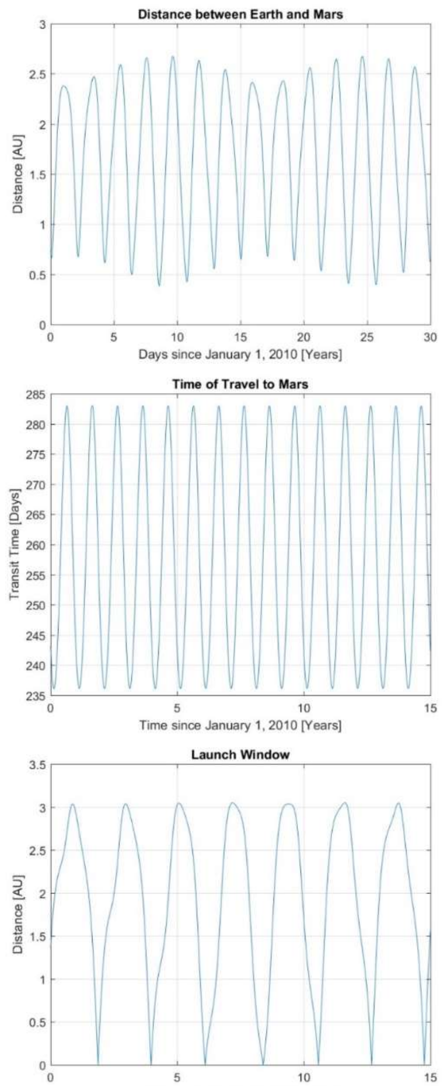


Figure 3: Top: Distance between Earth and Mars from 2010-2040. Middle: Time needed to reach Mars from 2010-2025. Bottom: Closest approach to Mars from 2010-2025. Generated by MATLAB ***Listing 1****.*

To send a spaceship from Earth to Mars involves the performance of a transfer orbit. The transfer orbit that requires the minimum amount of energy is known to be the Hohmann Transfer orbit, which allows a spaceship to efficiently transfer between non-intersection circular orbits, by connecting the periapsis to the apoapsis of an elliptical transfer ellipse. Since the orbits of Earth and Mars around the barycenter of the solar system are actually ellipsis that lie on different planes, the transfer problem becomes a more complicated and involved three-dimensional gravitational problem, involving many bodies. Typical Hohmann transfer orbits between Earth and Mars are began when Mars is $\sim 44^o$ ahead of Earth and take anywhere from $7-9$ months to complete a one-way trip.

The orbits of Earth and Mars were simulated with a MATLAB script from the year 2010 until 2040. Initial positions Earth and Mars were obtained from the JPL Horizons data (http://ssd.jpl.nasa.gov/horizons.cgi) for the date of January 1, 2010. Numerical integration was then used to calculate the accelerations, velocities, and positions in three-dimensions of the Earth and Mars as they orbited the barycenter of the solar system at each time step over the 30-year period. The distance between the two planets was then calculated at every time step and plotted in **Figure 3 (a)**. The transfer orbits were then calculated using an approximation that found the point on the Mars orbit exactly opposite from the current position of Earth. The time needed to get to that position was than roughly calculated with Kepler's Law ($P^2 = a^3$) and plotted in **Figure 3 (b)**. The travel times were found to fluctuate between $236\ days$ and $283\ days$. Finally, the resulting distance from Mars for each resulting transfer orbit was calculated and plotted to provide an approximation of the ideal launch windows to launch a spaceship to Mars from Earth. This is plotted in **Figure 3 (c)**.

The calculated launch windows are presented below with the accepted launch windows.

Table 2: Launch Window from Earth to Mars

Launch Dates	
Documented	**Calculated**
$November\ 8, 2011$	$November\ 15, 2011$
$December\ 31, 2013$	$December\ 19, 2013$
$March\ 21, 2016$	$February\ 7, 2016$
$May\ 17, 2018$	$May\ 23, 2018$
$July\ 18, 2020$	$July\ 28, 2020$
$September\ 14, 2022$	$September\ 4, 2022$
$October\ 5, 2024$	$October\ 3, 2024$

Launch window data from Earth to Mars were used from the following resource.
http://www.ltas-vis.ulg.ac.be/cmsms/uploads/File/InterplanetaryMissionDesignHandbook.pdf

The calculated launch windows matched well with the publicly available data and indicated that the MATLAB simulation provided reasonably accurate results. The MATLAB script also correctly predicted the date of closest approach from 2010-2040 between Mars and Earth to the correct date of July 31, 2018 and the correct distance of $0.3822\ AU$.

The next step was to calculate the necessary change in velocity of the spacecraft to enter the near Hohmann transfer orbit at a precise date, and to calculate the resulting trajectory. The $May\ 23, 2018$ launch date was chosen to simulate and calculate the trajectory for. This was done by calculating the current velocity of the Earth and calculating the velocity necessary to begin the transfer orbit. The orbiting speed at any point in an elliptical orbit is calculated as follows:

$$U = \sqrt{\frac{2\mu}{r} - \frac{\mu}{a}} \qquad \textbf{\textit{Eq. 9}}$$

Where $\mu = G * M$

At the launch location of Earth, the Earth's velocity was calculated to be $29.47 \frac{km}{s}$ and the necessary spaceship velocity to enter the transfer orbit was calculated from ***Eq. 9*** to be $32.44 \frac{km}{s}$, indicating that $\Delta v = 2.97 \frac{km}{s}$ was necessary to enter the transfer orbit before the spacecraft could accelerate only due to gravity to reach Mars. The resulting travel time was calculated to be $\sim 259\ days$ from Earth to Mars. The necessary adjustment in direction for the plane change was determined numerically and can be found in lines 157-165 of **Listing 2**. This was an approximate method, but produced a trajectory that completed the transfer orbit successfully. Time lapses of the trajectories are provided in **Figure 4** and **Figure 5**. It is noted that **Figure 3** and the resulting launch dates and closest approach were calculated with a time step of $180\ secon$, while **Figure 4** and **Figure 5** were generated with a time step of $3600\ seconds$ as is presented in **Listing 2**.

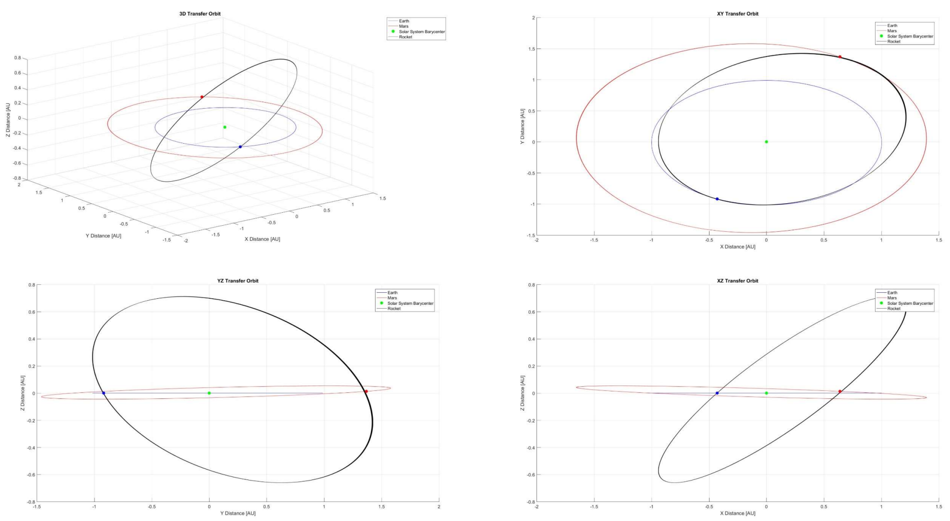


*Figure 4: Upper Right: Three-dimensional elliptical near Hohmann transfer orbit between Earth and Mars. Upper Left: View of the orbits in the XY-Plane. Lower Right: View of the orbits in the YZ-Plane. Lower Left: View of the orbits in the XZ-Plane. Generated by MATLAB **Listing 2**.*

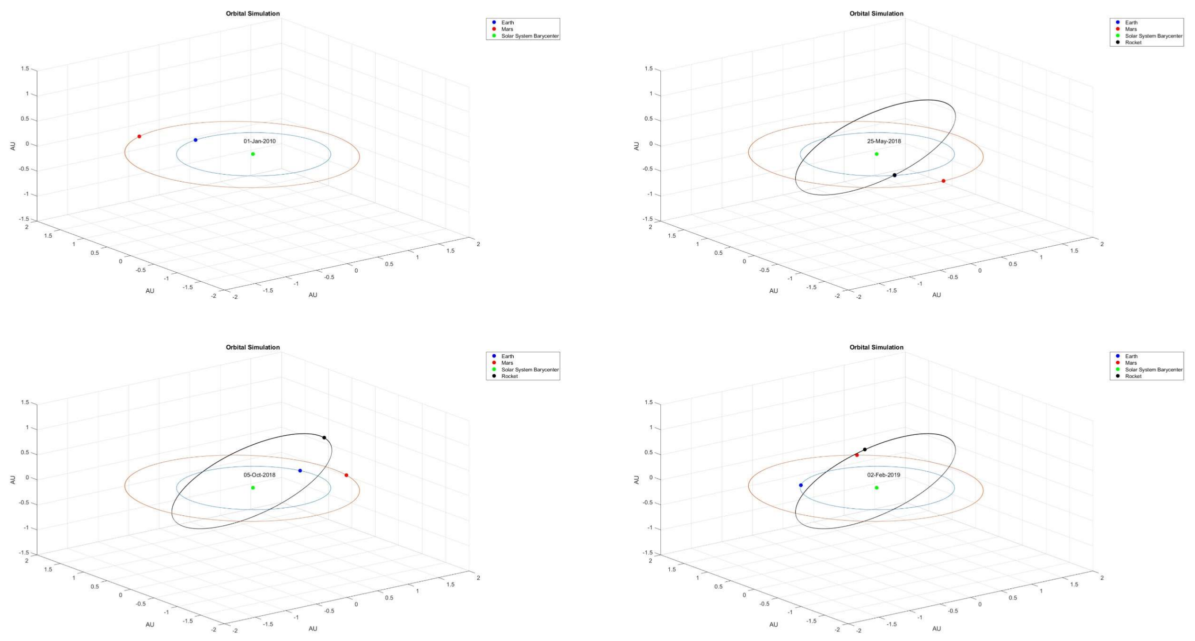


*Figure 5: Upper Right: Earth and Mars orbits about the solar system barycenter. Upper Left: Spaceship at launch from Earth. Lower Right: Spaceship in transit from Earth to Mars. Lower Right: Spaceship arriving at Mars. Generated by MATLAB **Listing 2**. More complete and accurate analytical calculations for this transfer trajectory can be found at resources such as http://www.braeunig.us/space.*

Elon Musk has stated that he hopes to reduce the travel time from the current $7 - 9\ mon\ \ hs$ to less than $80\ days$. This would involve continuously thrusting and further trajectory calculations that were out of the scope of this analysis. Further complications would arise from considering the velocity of the spacecraft when it reaches Mars, and the necessity to reduce the velocity to enter Mars orbit and eventually enter a parabolic orbit that would allow landing on Mars. Elon has expressed that capability of the Raptor engines to retrofire and slow down the spacecraft for touch down on Mars, but the capabilities introduce further complexities that SpaceX will need to consider to perform a successful mission. More precision in including the gravitational pull of other bodies and of avoiding high radiation areas would be of paramount importance when calculating the trajectory of the spacecraft, and would involve more computing resources than were available for this calculation.

CONCLUSIONS

The challenges regarding the Raptor engine to thrust the ITS from Earth to Mars and the trajectory of doing so in a reasonable amount of time for a crewed mission are only some of the challenges that

SpaceX faces before space travel to Mars becomes a reality. Elon's plan to send an unmanned mission by 2018 and a manned mission by 2024 are an ambitious timeline for a company that has yet to perform a manned mission, and substantial work and progress will be needed not only to produce the technologies necessary to make such a mission possible, but to manufacture the necessary hardware with the available financial resources. The creation of reusable rockets will provide a large step forward in producing the necessary hardware to reach Mars affordably, and as Elon has said, "So long as we continue to throw away rockets and spacecraft, we will never have true access to space" (p. 257, Vance). Other issues include the fact that the ideal launch window to travel to Mars only occurs once every 26 months leaves little room for error during launches, and all the hardware must be ready for launch without issue during the time frame. Since SpaceX has a reputation of "[experimenting] with new vehicles during actual launches in ways that other companies would dare not do," (p. 259, Vance) issues regarding the assurance of safety must be addressed thoroughly during manned launches and such experimentation would need to cease to be successful.

The simplified analysis performed and discussed above, it would appear to be possible to send a manned mission to Mars at some point. However, the probability of doing so within the next decade appears rather slim due to the large advancements in technology, mission planning, and manufacturing that still need to occur. As has been the case for all of Elon Musk's companies, he has set extremely ambitious goals and timelines, and rarely meets them promptly. However, his continued passion and work ethic have allowed his companies to deliver high quality products that are pushing innovation in the aerospace industry at a pace that has been much needed in the twenty-first-century. If Mr. Musk and the employees at his company continue to reduce the unprecedented problems presented by the ambitions of the company by just "[taking] it down to the physics," (p. 293, Vance) and problem solving, the goal of reaching Mars is attainable. Just not necessarily as soon as Elon Musk has expressed.

APPENDIX

Listing 1: MATLAB Code used for nozzle and thrust coefficient calculations and analysis.

Author: Jered Dominguez-Trujillo

Assign Variables

```
clear; clc; clf; close all;
WaterM = 18.01528;
WaterR = 8.3144598/WaterM; AirR = 0.2869;
g = 9.80665; Ttp = 216.65; Ptp = 22573; Pref = 101325; Tref = 288.15;
To = 3500; Po = 300*10^5 + Pref;
NozzleArea = 0.062912356383544;
% AeRatio = 200;
AeRatio = 40;
AExit = AeRatio * NozzleArea; AThroat = AExit / AeRatio;
```

Define Cp, Cv, and Gamma as functions of temperature

```
Cp = @(T) ((T >= 0.5 && T <= 1.7).*((30.092 + (6.832514.*T) +...
    (6.793435.*(T.^2)) - (2.53448.*(T.^3)) + (0.082139./(T.^2)))...
    ./ WaterM)) + ((T > 1.7 && T <= 6).*((41.96426 + (8.622053.*T) -...
    (1.49978.*(T.^2)) + (0.098119.*(T.^3)) + (-11.1576./(T.^2)))...
    ./ WaterM));
Cv = @(T) Cp(T) - WaterR;
Gamma = @(T) Cp(T)/Cv(T);
```

Establish A/A* as a function of Pe/Po and Me; Accommodating variation of Gamma (T)

```
set(gcf, 'Position', get(0, 'Screensize')); hold on
TempArray = [0.5, 1:1:6];
leg = cell(length(TempArray), 1);
count = 1;
for i = TempArray
    G = Gamma(i);
    leg{count} = ['Gamma = ' num2str(G)];
    Me = (1:0.001:6);
    A = (1./Me).*((2+(G - 1).*(Me.^2))./(G + 1)).^((G + 1)./(2.*(G-1)));
    count = count + 1;
    plot(A, Me)
end
title('$$\frac{A}{A*}\ as\ a\ Function\ of\ Mach\ Number$$',...
    'interpreter', 'latex', 'fontsize', 20)
ylabel('$$Mach\ Number$$', 'interpreter', 'latex', 'fontsize', 16)
xlabel('$$\frac{A}{A*}$$', 'interpreter', 'latex', 'fontsize', 16)
legend(leg, 'location', 'southeast')
set(findall(gcf,'Type','Line'),'Linewidth', 2); grid on
print('Area Ratio as Function of Mach Number', '-dpng')

close all; clf;
```

Calculating Design Exit Pressure and Altitude

```
criticalRatio = 300;
Pa = 200;
while criticalRatio > AeRatio
    Pa = Pa + 10;
    Temp = To / (((Po/Pa)^(Gamma(To/1000)-1)/Gamma(To/1000)));

    G = Gamma(Temp/1000);
    Me = 1; ARatioList = (1:0.025:1000);

    inx = 1;
    guess = (1/Me)*((2+(G-1)*Me^2)/(G+1))^((G+1)/(2*(G-1)));

    for ratio = ARatioList
        while guess < ratio
            Me = Me + 0.0025;
            guess = (1/Me)*((2+(G-1)*Me^2)/(G+1))^((G+1)/(2*(G-1)));
        end

        PRatio(inx) = 1/((1+((G-1)/2)*Me^2)^(G/(G-1)));
        Term1 = 2*G/(G-1);
        Term2 = (2/(G+1))^((G+1)/(G-1));
        Term3 = (1 - (PRatio(inx))^((G-1)/G));
        Term4 = sqrt(Term1 * Term2 * Term3);
        Cf(inx) = Term4 + (PRatio(inx) - Pa/Po) * ratio;
        inx = inx + 1;
    end

    criticalRatio = ARatioList(Cf==max(Cf));
end

height = ((1 - (Pa/Pref)^(1/5.2561))/(0.0065))*Tref;
```

Calculating Thrust

```
DesignThrust = Po * AThroat * max(Cf);
GroundThrust = DesignThrust - ((Pref - Pa)*AExit);
VacuumThrust = DesignThrust - ((0 - Pa)*AExit);
```

Printing Results

```
fprintf('Exit Temperature: %.2f K\n', Temp);
fprintf('Gamma: %.4f \n', G);
fprintf('Design Pressure: %.0f Pa\n', Pa);
fprintf('Design Altitude: %.2f m\n', height);
fprintf('Design Thrust: %.2f N\n', DesignThrust);
fprintf('Ground Thrust: %.2f N\n', GroundThrust);
fprintf('Vacuum Thrust: %.2f N\n', VacuumThrust);
fprintf('Pe/Po : %.6f \n', Pa/Po);
```

Plotting Thrust Coefficient of Design as a Function or A/A*

```
set(gcf, 'Position', get(0, 'Screensize')); plot(ARatioList, Cf, 'g'); hold on
TGamma = G;
for PaPo = [0.0001, 0.001, 0.002, 0.005, 0.01]
    Me = 1;
    Term = (1/Me)*((2+(TGamma-1)*Me^2)/(TGamma+1))^((TGamma+1)/(2*(TGamma-1)));
    inx = 1;
    for a = ARatioList
        while Term < a
            Me = Me + 0.01;
            Term = (1/Me)*((2+(TGamma-1)*Me^2)/(TGamma+1))^((TGamma+1)/(2*(TGamma-1)));
        end

        ratio = 1/((1+((TGamma-1)/2)*Me^2)^(TGamma/(TGamma-1)));

        Term1 = 2*TGamma/(TGamma-1);
        Term2 = (2/(TGamma+1))^((TGamma+1)/(TGamma-1));
        Term3 = (1-(ratio)^((TGamma-1)/TGamma));
        Term4 = sqrt(Term1*Term2*Term3);
        Cf(inx) = Term4+(ratio-(PaPo))*a;
        inx = inx + 1;
    end
    plot(ARatioList, Cf)
    axis tight
end
title('$$Thrust\ Coefficient\ as\ a\ Function\ of\ \frac{A}{A*}$$',...
    'interpreter', 'latex', 'fontsize', 20)
xlabel('$$\frac{A}{A*}$$', 'interpreter', 'latex', 'fontsize', 16)
ylabel('$$Thrust\ Coefficient$$', 'interpreter', 'latex', 'fontsize', 16)
legend({'Design ^{Pe}/_{Po} = .001674', '^{Pe}/_{Po} = 0.0001',...
    '^{Pe}/_{Po} = 0.001', '^{Pe}/_{Po} = 0.002', '^{Pa}/_{Po} = 0.005',...
    '^{Pe}/_{Po} = 0.01'}, 'location', 'southwest')
ylim([0 2])
set(findall(gcf,'Type','Line'),'LineWidth', 2)
grid on
print('Thrust Coefficient 40', '-dpng')
```

```
Area Ratio 40
Exit Temperature: 1377.06 K
Gamma: 1.2218
Design Pressure: 50380 Pa
Design Altitude: 5518.35 m
Design Thrust: 3076845.76 N
Ground Thrust: 2948642.96 N
Vacuum Thrust: 3203626.74 N
Pe/Po : 0.001674
```

```
Area Ratio 200
Exit Temperature: 936.39 K
Gamma: 1.2587
Design Pressure: 5250 Pa
Design Altitude: 19088.91 m
Design Thrust: 3167504.35 N
Ground Thrust: 1958643.42 N
Vacuum Thrust: 3233562.32 N
Pe/Po : 0.000175
```

Published with MATLAB® R2016a

Listing 2: MATLAB Code used to simulate Earth and Mars orbit and Hohmann Transfer between them.

Author: Jered Dominguez-Trujillo

Date: March 6, 2017

```
clear vars; close all;
set(gcf,'position',get(0,'screensize'))
```

Declare Variables

```
AU = 149597870700;
G = 6.674275935628303e-11;
MSolarSystem = 1.999*10^30;
RMars = 3.3899*10^6;
EarthYear = 365.24237; EarthDay = 86400;
dt = 3600; inx = 1;
StartYear = 2010; years = 30;
length = years * EarthYear * EarthDay;
dates = datetime(StartYear,01,01):caldays(1):datetime(StartYear+years+1,01,01);
t = zeros(1, ceil(length/dt));
t(inx) = 0;

axEarth = zeros(1, ceil(length/dt)); ayEarth = zeros(1, ceil(length/dt));
azEarth = zeros(1, ceil(length/dt));
vxEarth = zeros(1, ceil(length/dt)); vyEarth = zeros(1, ceil(length/dt));
vzEarth = zeros(1, ceil(length/dt));
xEarth = zeros(1, ceil(length/dt)); yEarth = zeros(1, ceil(length/dt));
zEarth = zeros(1, ceil(length/dt)); rEarth = zeros(1, ceil(length/dt));
alphaEarth = zeros(1, ceil(length/dt)); betaEarth = zeros(1, ceil(length/dt));
gammaEarth = zeros(1, ceil(length/dt));

axMars = zeros(1, ceil(length/dt)); ayMars = zeros(1, ceil(length/dt));
azMars = zeros(1, ceil(length/dt));
vxMars = zeros(1, ceil(length/dt)); vyMars = zeros(1, ceil(length/dt));
vzMars = zeros(1, ceil(length/dt));
xMars = zeros(1, ceil(length/dt)); yMars = zeros(1, ceil(length/dt));
zMars = zeros(1, ceil(length/dt)); rMars = zeros(1, ceil(length/dt));
alphaMars = zeros(1, ceil(length/dt)); betaMars = zeros(1, ceil(length/dt));
gammaMars = zeros(1, ceil(length/dt));

axRocket = zeros(1, ceil(length/dt)); ayRocket = zeros(1, ceil(length/dt));
azRocket = zeros(1, ceil(length/dt));
vxRocket = zeros(1, ceil(length/dt)); vyRocket = zeros(1, ceil(length/dt));
vzRocket = zeros(1, ceil(length/dt));
xRocket = zeros(1, ceil(length/dt)); yRocket = zeros(1, ceil(length/dt));
zRocket = zeros(1, ceil(length/dt)); rRocket = zeros(1, ceil(length/dt));
alphaRocket = zeros(1, ceil(length/dt)); betaRocket = zeros(1, ceil(length/dt));
gammaRocket = zeros(1, ceil(length/dt));
```

Data Gathered for Earth and Mars for January 1, 2010 from JPL

```
axEarth(inx) = 0; ayEarth(inx) = 0; azEarth(inx) = 0;
vxEarth(inx) = -2.978405751621624e+04; vyEarth(inx) = -5.451137243323289e+03;
vzEarth(inx) = 1.551580077431058;
xEarth(inx) = -2.689245210784379e+10; yEarth(inx) = 1.451618583042868e+11;
zEarth(inx) = -2.608728054337204e+06;
rEarth(inx) = sqrt(xEarth(inx)^2+yEarth(inx)^2+zEarth(inx)^2);
alphaEarth(inx) = acos(xEarth(inx)/rEarth(inx)); betaEarth(inx) = acos(yEarth(inx)/rEarth(inx));
gammaEarth(inx) = acos(zEarth(inx)/rEarth(inx));
axMars(inx) = 0; ayMars(inx) = 0; azMars(inx) = 0;
vxMars(inx) = -2.074332689615267e+04; vyMars(inx) = -8.817501559049284e+03;
vzMars(inx) = 3.248028553828797e+02;
xMars(inx) = -1.097178350768191e+11; yMars(inx) = 2.179958733988708e+11;
zMars(inx) =  7.239687840140940e+09;
rMars(inx) = sqrt(xMars(inx)^2+yMars(inx)^2+zMars(inx)^2);
alphaMars(inx) = acos(xMars(inx)/rMars(inx)); betaMars(inx) = acos(yMars(inx)/rMars(inx));
gammaMars(inx) = acos(zMars(inx)/rMars(inx));
distanceEarthtoMars = zeros(1, ceil(length/dt));
distanceEarthtoMars(inx) = sqrt((xMars(inx)-xEarth(inx))^2+(yMars(inx)-
yEarth(inx))^2+(zMars(inx)-zEarth(inx))^2);
```

Calculate Earth and Mars Trajectories

```
while t(inx) < length
    inx = inx + 1;
    alphaEarth(inx)=acos(xEarth(inx-1)/rEarth(inx-1));
    betaEarth(inx)=acos(yEarth(inx-1)/rEarth(inx-1));
    gammaEarth(inx) = acos(zEarth(inx-1)/rEarth(inx-1));
    rEarth(inx) = sqrt(xEarth(inx-1)^2+yEarth(inx-1)^2+zEarth(inx-1)^2);
    axEarth(inx) = -(G*MSolarSystem)/(rEarth(inx)^2)*cos(alphaEarth(inx));
    ayEarth(inx) = -(G*MSolarSystem)/(rEarth(inx)^2)*cos(betaEarth(inx));
    azEarth(inx) = -(G*MSolarSystem)/(rEarth(inx)^2)*cos(gammaEarth(inx));
    vxEarth(inx)=vxEarth(inx-1)+axEarth(inx)*dt; vyEarth(inx)=vyEarth(inx-1)+ayEarth(inx)*dt;
    vzEarth(inx)=vzEarth(inx-1)+azEarth(inx)*dt;
    xEarth(inx)=xEarth(inx-1)+vxEarth(inx)*dt; yEarth(inx)=yEarth(inx-1)+vyEarth(inx)*dt;
    zEarth(inx)=zEarth(inx-1)+vzEarth(inx)*dt;
    alphaMars(inx)=acos(xMars(inx-1)/rMars(inx-1));betaMars(inx)=acos(yMars(inx-1)/rMars(inx-1));
    gammaMars(inx) = acos(zMars(inx-1)/rMars(inx-1));
    rMars(inx) = sqrt(xMars(inx-1)^2+yMars(inx-1)^2+zMars(inx-1)^2);
    axMars(inx) = -(G*MSolarSystem)/(rMars(inx)^2)*cos(alphaMars(inx));
    ayMars(inx) = -(G*MSolarSystem)/(rMars(inx)^2)*cos(betaMars(inx));
    azMars(inx) = -(G*MSolarSystem)/(rMars(inx)^2)*cos(gammaMars(inx));
    vxMars(inx)=vxMars(inx-1)+axMars(inx)*dt; vyMars(inx)=vyMars(inx-1)+ayMars(inx)*dt;
    vzMars(inx)=vzMars(inx-1)+azMars(inx)*dt;
    xMars(inx)=xMars(inx-1)+vxMars(inx)*dt; yMars(inx) = yMars(inx-1) + vyMars(inx) * dt;
    zMars(inx) = zMars(inx-1) + vzMars(inx) * dt;
    distanceEarthtoMars(inx) = sqrt((xMars(inx)-xEarth(inx))^2+(yMars(inx)-
yEarth(inx))^2+(zMars(inx)-zEarth(inx))^2);

    t(inx) = t(inx-1) + dt;
end
```

Find Time and Distance of Closest Approach

```
closestApproach = find(distanceEarthtoMars==min(distanceEarthtoMars))*dt/EarthDay/EarthYear;
ApproachYear = StartYear + floor(closestApproach);
ApproachDay = floor(rem(closestApproach, 1) * EarthYear);
datesApproach = datetime(ApproachYear,01,01):caldays(1):datetime(ApproachYear,12,31);
fprintf('Mars Closest Approach to Earth between %s and %s occurs on %s at a distance of %.4f
AU\n',...
    num2str(StartYear), num2str(StartYear + years), datestr(datesApproach(ApproachDay)), ...
    min(distanceEarthtoMars)/AU);
```

Calculating Transit Time and Target Distance if Launched at Each Time Step

```
alphaEarth = round(atan2(real(yEarth), real(xEarth)), 3);
alphaMars = round(atan2(real(yMars), real(xMars)), 3);
TransitTime = zeros(1, ceil(inx)); TargetDistance = zeros(1, ceil(inx));
for ii = 1:0.5*inx
    if alphaEarth(ii) > 0
        value = alphaEarth(ii) - round(pi, 3);
    else
        value = alphaEarth(ii) + round(pi, 3);
    end
    MarsOppositeInx = find(abs(alphaMars - value)<0.0001);
    if isempty(MarsOppositeInx) == 0
        d = sqrt((xEarth(ii) - xMars(MarsOppositeInx(1)))^2+(yEarth(ii) -
yMars(MarsOppositeInx(1)))^2+(zEarth(ii) - zMars(MarsOppositeInx(1)))^2);
        TransitTime(ii) = ((sqrt((((d/AU))/2)^3))/2)*EarthYear;
        TimeinSeconds=TransitTime(ii)*EarthDay; TimeSteps=real(ceil(TimeinSeconds/dt));
        TargetDistance(ii) = sqrt(((xMars(MarsOppositeInx(1)) - xMars(ii+TimeSteps))^2+...
            (yMars(MarsOppositeInx(1)) - yMars(ii+TimeSteps))^2+...
            (zMars(MarsOppositeInx(1)) - zMars(ii+TimeSteps))^2));
    else
        TransitTime(ii) = TransitTime(ii - 1); TargetDistance(ii) = TargetDistance(ii - 1);
    end
end
```

Print Appropriate Launch Dates

```
LaunchDates = find(TargetDistance(1:floor(0.5*inx))/AU < 40*RMars/AU);
datecount = size(LaunchDates);
for ii = 1:datecount(2)
    Val = LaunchDates(ii)*dt/EarthDay/EarthYear;
    LaunchHour = round(rem(LaunchDates(ii)/24, 1) * 24, 0);
    LaunchYear = StartYear + floor(Val); LaunchDay = floor(rem(Val, 1) * EarthYear);
    datesLaunch = datetime(LaunchYear,01,01):caldays(1):datetime(LaunchYear,12,31);
    fprintf('Launch to Mars: %s. Closest Approach to Mars: %.4f Mars Radii.  Travel Time: %.4f
Days\n',...
        datestr(datesLaunch(LaunchDay)), TargetDistance(LaunchDates(ii))/RMars, ...
        TransitTime(LaunchDates(ii)));
end
```

Calculate Rocket Start and End Positions

```
LaunchofInterest = LaunchDates(end);  % 2018 Launch
if alphaEarth(LaunchofInterest) > 0
    val = alphaEarth(LaunchofInterest) - round(pi, 3);
else
    val = alphaEarth(LaunchofInterest) + round(pi, 3);
end
MarsOppositeInx = find(abs(alphaMars - val)<0.0001);
dEtoM = sqrt(((xEarth(LaunchofInterest)-xMars(MarsOppositeInx(1)))^2)+...
    ((yEarth(LaunchofInterest)-yMars(MarsOppositeInx(1)))^2)+...
    ((zEarth(LaunchofInterest)-zMars(MarsOppositeInx(1)))^2));
Vp = sqrt(G*MSolarSystem*((2/rEarth(LaunchofInterest))-(1/(dEtoM/2))));
```

Initial Rocket Parameters Calculated to Complete Transfer Orbit

```
inx = 1; t = zeros(1, ceil(length/dt)); t(inx) = 0; rad = 0 ; vtest = 100000;
axRocket(inx) = 0; ayRocket(inx) = 0; azRocket(inx) = 0;
while vtest > Vp
    vxRocket(inx) = ((-1*vxEarth(LaunchofInterest))/abs(vxEarth(LaunchofInterest)))*...
        (-abs(vxEarth(LaunchofInterest)) + (Vp-abs(vxEarth(LaunchofInterest)))*(1-cos(rad)));
    vyRocket(inx) = ((-1*vyEarth(LaunchofInterest))/abs(vyEarth(LaunchofInterest)))*...
        (abs(vyEarth(LaunchofInterest))+(Vp+abs(vyEarth(LaunchofInterest)))*cos(pi-rad));
    vzRocket(inx) = ((vzEarth(LaunchofInterest))/abs(vzEarth(LaunchofInterest)))*...
        (abs(vzEarth(LaunchofInterest))-(Vp-abs(vzEarth(LaunchofInterest)))*cos(rad));
    vtest = sqrt(vxRocket(inx)^2 + vyRocket(inx)^2 + vzRocket(inx)^2); rad = rad + 0.00001;
end
xRocket(inx) = xEarth(LaunchofInterest); yRocket(inx) = yEarth(LaunchofInterest);
zRocket(inx) =  zEarth(LaunchofInterest);
rRocket(inx) = sqrt(xRocket(inx)^2+yRocket(inx)^2+zRocket(inx)^2);
alphaRocket(inx)=acos(xRocket(inx)/rRocket(inx));betaRocket(inx)=acos(yRocket(inx)/rRocket(inx));
gammaRocket(inx) = acos(zRocket(inx)/rRocket(inx));
```

Calculate Rocket Trajectory

```
while t(inx) < length
    inx = inx + 1;
    alphaRocket(inx) = acos(xRocket(inx-1)/rRocket(inx-1));
    betaRocket(inx) = acos(yRocket(inx-1)/rRocket(inx-1));
    gammaRocket(inx) = acos(zRocket(inx-1)/rRocket(inx-1));
    rRocket(inx) = sqrt(xRocket(inx-1)^2+yRocket(inx-1)^2+zRocket(inx-1)^2);
    axRocket(inx) = -(G*MSolarSystem)/(rRocket(inx)^2)*cos(alphaRocket(inx));
    ayRocket(inx) = -(G*MSolarSystem)/(rRocket(inx)^2)*cos(betaRocket(inx));
    azRocket(inx) = -(G*MSolarSystem)/(rRocket(inx)^2)*cos(gammaRocket(inx));
    vxRocket(inx) = vxRocket(inx-1) + axRocket(inx) * dt;
    vyRocket(inx) = vyRocket(inx-1) + ayRocket(inx) * dt;
    vzRocket(inx) = vzRocket(inx-1) + azRocket(inx) * dt;
    xRocket(inx)=xRocket(inx-1)+vxRocket(inx)*dt; yRocket(inx)=yRocket(inx-1)+vyRocket(inx)*dt;
    zRocket(inx) = zRocket(inx-1) + vzRocket(inx) * dt;
    t(inx) = t(inx - 1) + dt;
end
```

Plot Animation from 2010 Until Arrival at Mars

```
flag =false;
for ii = 1:LaunchofInterest + (real(max(TransitTime))*EarthDay/dt)
     if rem(t(ii), EarthDay) == 0 && t(ii) > 0
         clf; hold on; view(3)
         plot3(real(xEarth(ii))/AU, real(yEarth(ii))/AU, real(zEarth(ii))/AU, 'bo', ...
             'MarkerFaceColor','b')
         plot3(real(xMars(ii))/AU, real(yMars(ii))/AU, real(zMars(ii))/AU, 'ro', ...
             'MarkerFaceColor','r')
         plot3(0, 0, 0, 'go', 'MarkerFaceColor','g')
         if ii >= LaunchofInterest
             plot3(real(xRocket(ii-LaunchofInterest))/AU, real(yRocket(ii-
LaunchofInterest))/AU,...
                 real(zRocket(ii-LaunchofInterest))/AU, 'ko', 'MarkerFaceColor', 'k')
             plot3(real(xRocket)/AU, real(yRocket)/AU, real(zRocket)/AU, 'k', 'linewidth', 0.25)
             flag = true;
         end
         plot3(real(xEarth/AU), real(yEarth/AU), real(zEarth/AU))
         plot3(real(xMars/AU), real(yMars/AU), real(zMars/AU))
         xlim([-2, 2]); ylim([-2, 2]); zlim([-1.5, 1.5])
         xlabel('AU'); ylabel('AU'); zlabel('AU')
         title('Orbital Simulation')
         text(0, 0.2, 0.2, datestr(dates(ceil(t(ii)/EarthDay))));
         if flag == true
            legend({'Earth', 'Mars', 'Solar System Barycenter', 'Rocket'})
         else
             legend({'Earth', 'Mars', 'Solar System Barycenter'})
         end
         grid on
         print(['Simulation\Time' num2str(ii)], '-dpng')
         pause(0.01)
     end
end
```

Plot and Save Figures

```
clf; close all;
set(gcf,'position',get(0,'screensize'))
figure(1); hold on; view(3);
plot3(xEarth/AU, yEarth/AU, zEarth/AU, 'b');
plot3(xMars/AU, yMars/AU, zMars/AU, 'r'); plot3(0, 0, 0, 'go', 'MarkerFaceColor','g')
plot3(xRocket/AU, yRocket/AU, zRocket/AU, 'k');
plot3(xEarth(LaunchofInterest)/AU, yEarth(LaunchofInterest)/AU, zEarth(LaunchofInterest)/AU,
'bo', 'MarkerFaceColor', 'b');
plot3(xMars(MarsOppositeInx(1))/AU, yMars(MarsOppositeInx(1))/AU, zMars(MarsOppositeInx(1))/AU,
'ro', 'MarkerFaceColor', 'r');
legend({'Earth', 'Mars', 'Solar System Barycenter', 'Rocket'})
title('3D Transfer Orbit');
xlabel('X Distance [AU]'); ylabel('Y Distance [AU]'); zlabel('Z Distance [AU']); grid on;
print('3D Transfer Orbit', '-dpng');
```

```
figure(2); set(gcf,'position',get(0,'screensize')); hold on; plot(xEarth/AU, yEarth/AU, 'b');
plot(xMars/AU, yMars/AU, 'r');
plot(0, 0, 'go', 'MarkerFaceColor', 'g'); plot(xRocket/AU, yRocket/AU, 'k');
plot(xEarth(LaunchofInterest)/AU, yEarth(LaunchofInterest)/AU, 'bo', 'MarkerFaceColor', 'b')
plot(xMars(MarsOppositeInx(1))/AU, yMars(MarsOppositeInx(1))/AU, 'ro', 'MarkerFaceColor', 'r')
legend({'Earth', 'Mars', 'Solar System Barycenter', 'Rocket'})
title('XY Transfer Orbit');
xlabel('X Distance [AU]'); ylabel('Y Distance [AU]'); grid on;
print('XY Transfer Orbit', '-dpng');

figure(3); set(gcf,'position',get(0,'screensize')); hold on; plot(yEarth/AU, zEarth/AU, 'b');
plot(yMars/AU, zMars/AU, 'r');
plot(0, 0, 'go', 'MarkerFaceColor', 'g'); plot(yRocket/AU, zRocket/AU, 'k');
plot(yEarth(LaunchofInterest)/AU, zEarth(LaunchofInterest)/AU, 'bo', 'MarkerFaceColor', 'b')
plot(yMars(MarsOppositeInx(1))/AU, zMars(MarsOppositeInx(1))/AU, 'ro', 'MarkerFaceColor', 'r')
legend({'Earth', 'Mars', 'Solar System Barycenter', 'Rocket'})
title('YZ Transfer Orbit');
xlabel('Y Distance [AU]'); ylabel('Z Distance [AU]'); grid on;
print('YZ Transfer Orbit', '-dpng');

figure(4); set(gcf,'position',get(0,'screensize')); hold on; plot(xEarth/AU, zEarth/AU, 'b');
plot(xMars/AU, zMars/AU, 'r');
plot(0, 0, 'go', 'MarkerFaceColor', 'g'); plot(xRocket/AU, zRocket/AU, 'k');
plot(xEarth(LaunchofInterest)/AU, zEarth(LaunchofInterest)/AU, 'bo', 'MarkerFaceColor', 'b')
plot(xMars(MarsOppositeInx(1))/AU, zMars(MarsOppositeInx(1))/AU, 'ro', 'MarkerFaceColor', 'r')
legend({'Earth', 'Mars', 'Solar System Barycenter', 'Rocket'})
title('XZ Transfer Orbit');
xlabel('X Distance [AU]'); ylabel('Z Distance [AU]'); grid on;
print('XZ Transfer Orbit', '-dpng');

figure(5);
plot(t(1:inx)/EarthDay/EarthYear, real(distanceEarthtoMars(1:inx)) / AU);
ylim([0, 3]);
xlabel('Days since January 1, 2010 [Years]'); ylabel('Distance [AU]');
title('Distance between Earth and Mars');
xlim([0 years]); grid on;
print('Distance Earth and Mars', '-dpng');

figure(6);
plot(t(1:floor(0.5*inx))/EarthDay/EarthYear, real(TargetDistance(1:floor(0.5*inx)))/AU);
xlabel('Time since January 1, 2010 [Years]'); ylabel('Distance [AU]');
title('Launch Window'); grid on;
print('Launch Window', '-dpng');

figure(7);
plot(t(1:floor(0.5*inx))/EarthDay/EarthYear, real(TransitTime(1:floor(0.5*inx))));
xlabel('Time since January 1, 2010 [Years]'); ylabel('Transit Time [Days]');
title('Time of Travel to Mars'); grid on;
print('Time of Travel', '-dpng');
```

```
Mars Closest Approach to Earth between 2010 and 2040 occurs on 01-Aug-2018 at a distance of 0.3825 AU

Launch to Mars: 15-Nov-2011. Closest Approach to Mars: 5.8623 Mars Radii.  Travel Time: 259.9553 Days
Launch to Mars: 15-Nov-2011. Closest Approach to Mars: 19.8462 Mars Radii.  Travel Time: 259.9542 Days
Launch to Mars: 15-Nov-2011. Closest Approach to Mars: 31.0971 Mars Radii.  Travel Time: 259.9280 Days
Launch to Mars: 19-Dec-2013. Closest Approach to Mars: 5.7588 Mars Radii.  Travel Time: 246.3986 Days
Launch to Mars: 19-Dec-2013. Closest Approach to Mars: 27.4071 Mars Radii.  Travel Time: 246.3831 Days
Launch to Mars: 19-Dec-2013. Closest Approach to Mars: 5.7687 Mars Radii.  Travel Time: 246.3830 Days
Launch to Mars: 08-Feb-2016. Closest Approach to Mars: 30.8875 Mars Radii.  Travel Time: 236.3355 Days
Launch to Mars: 08-Feb-2016. Closest Approach to Mars: 30.8637 Mars Radii.  Travel Time: 236.3337 Days
Launch to Mars: 23-May-2018. Closest Approach to Mars: 14.5713 Mars Radii.  Travel Time: 258.6647 Days
Launch to Mars: 23-May-2018. Closest Approach to Mars: 14.5662 Mars Radii.  Travel Time: 258.6825 Days
Launch to Mars: 23-May-2018. Closest Approach to Mars: 14.5587 Mars Radii.  Travel Time: 258.7097 Days
Launch to Mars: 23-May-2018. Closest Approach to Mars: 37.3640 Mars Radii.  Travel Time: 258.7360 Days
Launch to Mars: 24-May-2018. Closest Approach to Mars: 11.4190 Mars Radii.  Travel Time: 258.7368 Days
Launch to Mars: 24-May-2018. Closest Approach to Mars: 11.4202 Mars Radii.  Travel Time: 258.7546 Days
Launch to Mars: 24-May-2018. Closest Approach to Mars: 37.3638 Mars Radii.  Travel Time: 258.7996 Days
```

Published with MATLAB® R2016a