MTP2 Praktikumsbericht

Karolina Bernat, Olliver Steenbuck

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

1 Weltraummissionen

1.1 Erdumkreisung, Fluchtgeschwindigkeit und geostationäre Bahn

Im Folgenmden wird beschrieben, wie eine antribslose Phase eines Satelliten, der mit einer Trägerrakete in eine Startposition x_0 gebracht wird und dann antrieblos weiter fliegt. Dabei ist der Einfluss des Satelliten auf die Erde zu vernachlässigen.

Im Modell folgende Konstanten sind bekannt:
Erdradius: $r_E = 6378km$
Erdmasse: $m_E = 5.9736 \cdot 10^{24}kg$
Gravitationskonstante: $G = 66.743 \cdot 10^{-12} \frac{m^3}{kg \cdot s^2}$

Startpositionsvektor

Zur Errechnung der Startposition des Satellitens nutzen wir:

$$\overrightarrow{x_0} = [\cos(\delta \cdot (r + h_0)), \sin(\delta \cdot (r + h_0))]$$

Die Funktion ***"Startposition"*** sieht folgendermaßen aus:

```
function x0 = Startposition(deltaDegree, h0)

r = 6378000;   % Erdradius [m]
h = r + h0;    % Hypothenuse
x0 = [cosd(deltaDegree) * h ; sind(deltaDegree) * h ];
```


```
function x0 = Startposition(deltaDegree , h0)

r = 6378000;   % Erdradius [m]
h = r + h0;    % Hypothenuse
x0 = [cosd(deltaDegree) * h ; sind(deltaDegree) * h ];
```

Startgeschwindigkeitsvektor

Im nächsten Schritt wird der Startgeschwindigkeitswektor $\overrightarrow{v_{0,Welt}}$ berechnet. Hierfür werden zunächst die Einheitsvektoren in Tangential- sowie Normalrichtung konstruiert. Danach werden die Tangential- und Normalkomponente der Startgeschwindigkeit berechnet, wobei daraus die Startgeschwindigkeit zusammengebaust wird. Die Funktion ***"vStart"***, die diese Schritte durchführt, sieht folgendermaßen aus:

```
function v0welt = vStart(v0, thetaDegree, x0)

% v0 - Startgeschwindigkeit
% thetaDegree Winket theta in Grad
% x0 STartposition

nE = x0 / norm(x0);      % Einheitsvektor in Normalrichtung fuer x0
tE = [nE(2) ; -nE(1)];   % Einheitvektor in Tangentialrichtung fuer x0

vt = cosd(thetaDegree) * v0;     % Tangentialkomponente fuer v0
vn = sind(thetaDegree) * v0;     % Normalkomponente fuer v0

v0welt = tE * vt + nE * vn;      % Geschwindigkeitsvektor
```

Beschleunigung

Zur Berechnung der Beschleunigung des Satellitens wird die stets aktuelle Position des Satellitens benötigt. Durch die Summierung der Kräfte:

$$\sum F = m \cdot a$$

ergibt sich:

$$a = \frac{\sum F}{m}$$

Dabei ist m die Masse des Satellitens, die hier vernachlässigt werden kann. Somit ergibt sich:

$$a = F_{SE}$$

und folglich:

$$a = G \cdot \frac{m_E \cdot m_S}{r_{SE}^2} \cdot \overrightarrow{e_{SE}}$$

wobei F_{SE} die Kraft die vom Satelliten zu Erde wirkt, r_{SE} der Abstand von der Erdmitte zum Satelliten und $\overrightarrow{e_{SE}}$ der Einheitsvektor vom Satelliten zur Erde ist und m_S (Masse des Satellitens) vernachlässigt wird.

Die Funktion ***"Beschleunigung"*** sieht folgendermaßen aus:

```
function a = Beschleunigung(xAktuell)

xE = xAktuell / norm(xAktuell);   % Einheitsvektor fuer xAktuell
eSE = xE * (-1);    % Einheitsvektor vom Satelliten zu Erde - umgekehrte Richtung zu xE

r = norm(xAktuell); % Abstand von Erdmitte zum Satellit

mE = 5.9736 * 10^24;    % Erdmasse in [kg]
G = 66.743 * 10^-12;    % Gravitationskonstante in [m^3/kg*s^2]

a = G * mE / r^2 * eSE; % Beschleunigung
```

Kontakt

Sobald der Satellit die Erde berührt soll die Simulation beendet werden. Hierfür wurde die Funktion ***“Kontakt”*** geschrieben, die dies kontrolliert und wie folgt ausschaut:

```
function endBedingung = Kontakt(xAktuell)

rErde = 6378000;
xAbstand = norm(xAktuell);

if (xAbstand > rErde)
        endBedingung = 0;
else
    endBedingung = 1;
end
```

Gesamtmodell

Das vollständige Modell wurde in Simulink modelliert und kann der folgenden Abbildung entnommen werden:

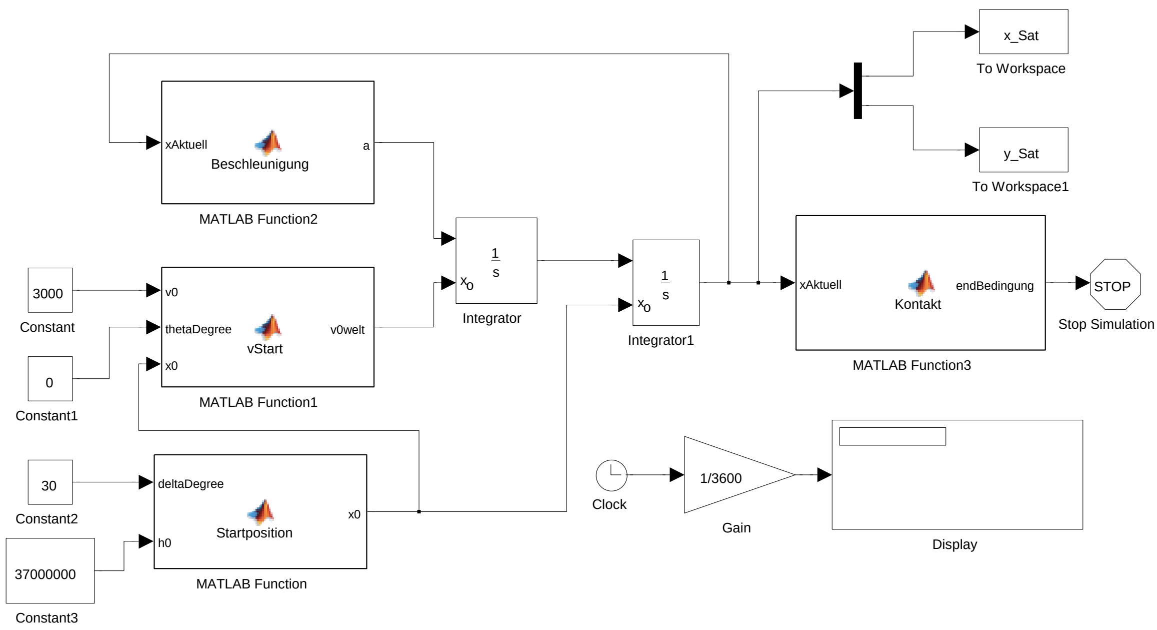

Abbildung 1.1: Gesamtmodell in Matlab/Simulink

Versuchsdurchführung

- Eine Kreisbahn in gleicher Höhe

Folgende Voreinstellungen wurden für die Simulation gewählt:

$$\delta = 30^{\circ},\, h_0 = 400km,\, \theta = 0^{\circ}$$

Bei einer Startgeschwindigkeit $v_0 = 7{,}65\frac{km}{s}$ und der Simulationszeit von $1{,}542h$ umkreist der Satellit die Erde bei einer konstanten Höhe genau ein Mal wie in der folgenden Abbildung zu sehen ist:

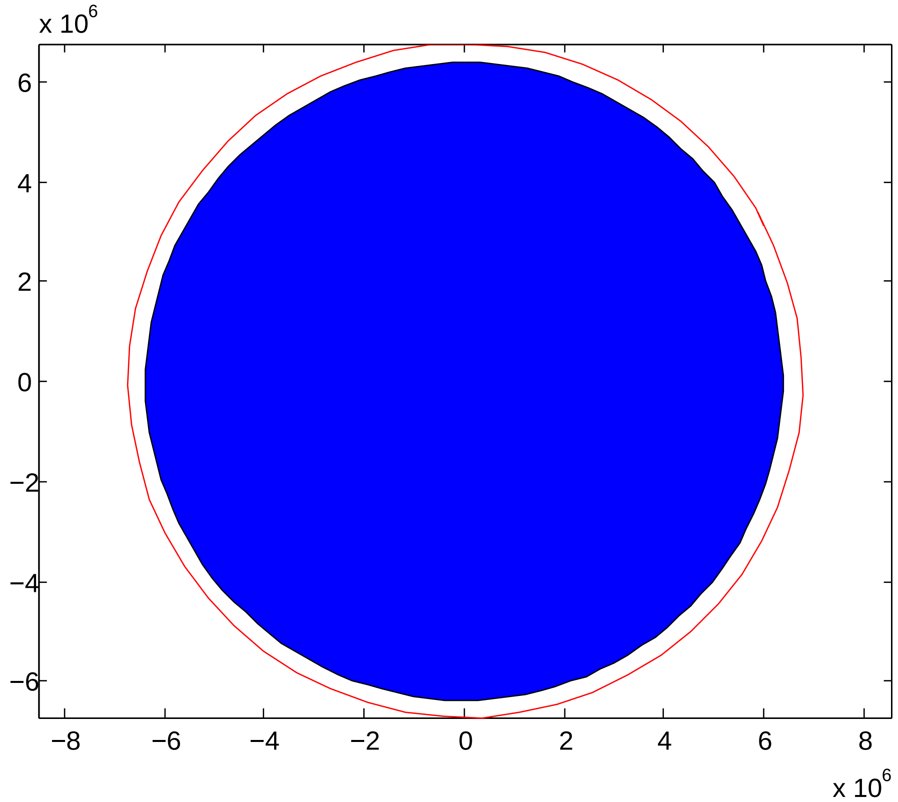

Abbildung 1.2: Eine Kreisbahn in gleicher Höhe

- Entfliehen der Erde

Folgende Voreinstellungen wurden für die Simulation gewählt:

$$\delta = 30^\circ,\, h_0 = 400km,\, \theta = 0^\circ$$

Bei einer Startgeschwindigkeit $v_0 = 10{,}85\frac{km}{s}$ und der Simulationszeit von $1.000.000s$ entflieht der Satellit der Erde, wie in der nachfolgenden Grafik zu sehen ist:

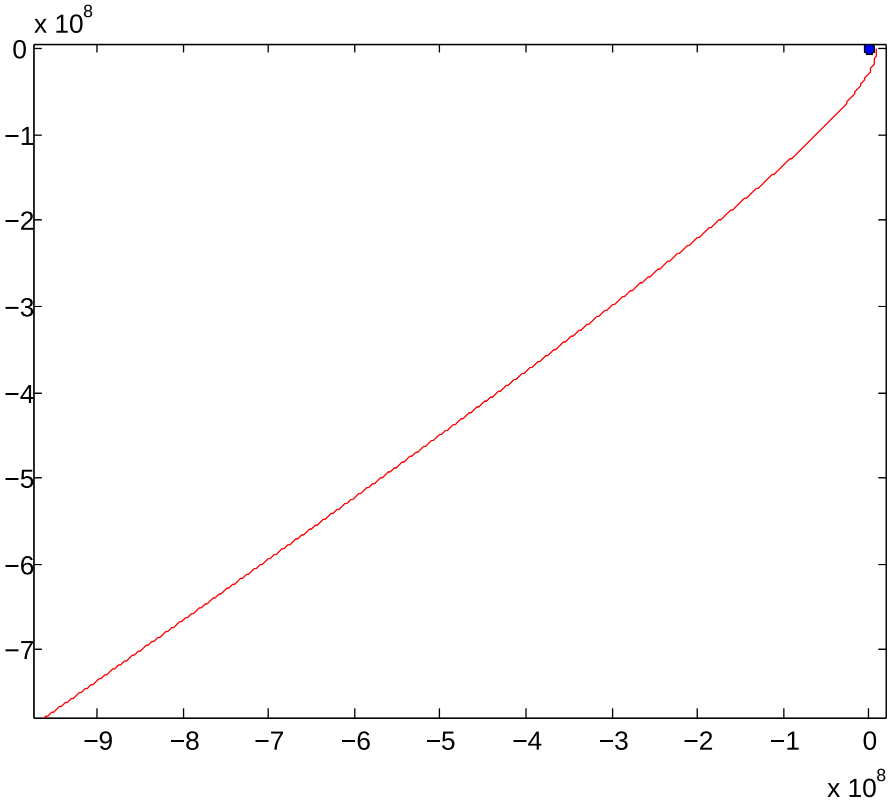

Abbildung 1.3: Entfliehen der Erde

- Eine Kreisbahn innerhalb eines Tages

$$\delta = 30^{\circ}, \theta = 0^{\circ}$$

Bei einer Starthöhe von $h_0 = 37.000km$ und der Startgeschwindigkeit von $v_0 = 2{,}995\frac{km}{s}$ umkreist der Satellit die Erde ein Mal bei der Simulationszeit von $24h$, wie in der nachfolgenden Grafik zu sehen ist:

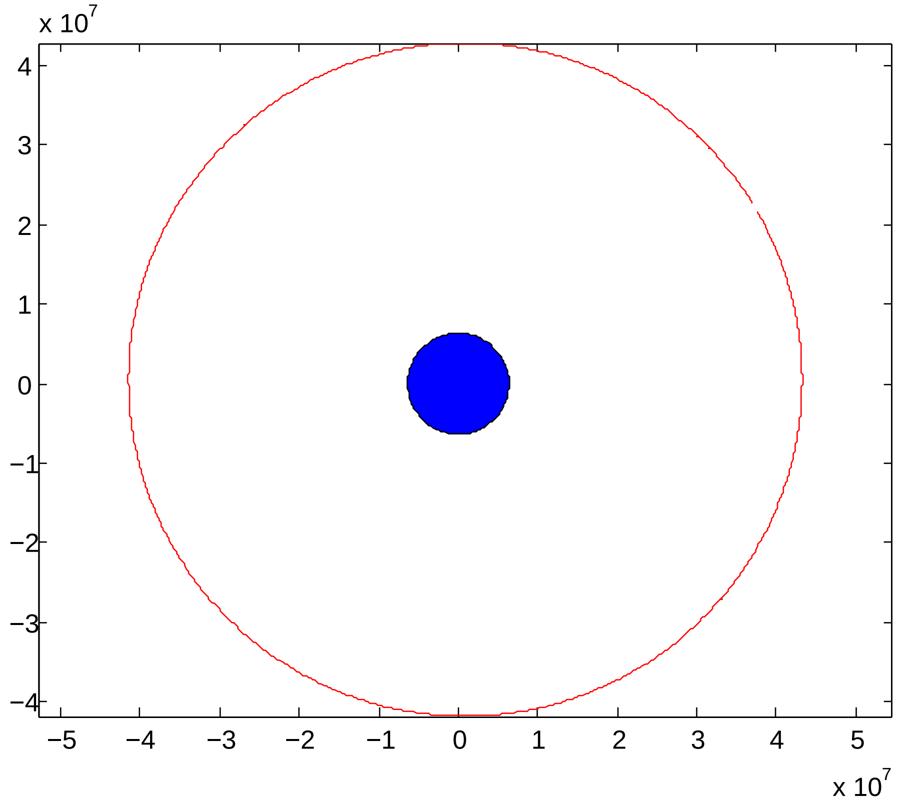

Abbildung 1.4: Eine Kreisbahn innerhalb eines Tages

1.2 Mondumkreisung

In diesem Abschnitt wird die Simulation eines Satelliten, der von der Erde zum Mond fliegt, vorgestellt. Hierfür wird angenommen, dass sich der Mond nicht bewegt.

Folgende Kennzehalen sind bekannt:
Mondposition (fest): $x_M = (0, -380.000)^T km$
Mondmasse: $m_M = 7{,}3480 \cdot 10^{22} kg$

Beschleunigung

Für das Matlab-Modell wird die Funktion ***"Beschleunigung"*** ergänzt, dass die vom Mond auf den Satelliten wirkende Kraft F_M berücksichtigt wird. Diese wird wie folgt berechnet:

$$F_{SE} = G \cdot \frac{m_E}{r_{SE}^2} \cdot \overrightarrow{e_{SE}}$$

$$F_{SM} = G \cdot \frac{m_M}{r_{SM}^2} \cdot \overrightarrow{e_{SM}}$$

wobei m_M die Mondmasse, r_{SM} die Entfernung vom Satelliten zu Mond und $\overrightarrow{e_{SM}}$ der Einheitsvektor vom Satelliten zu Mond ist.

Um die BEschleunigung des Satelliten zu errechnen nutzen wir:

$$\sum F = m \cdot a$$

wobei m die Masse des Satelliten ist und im Modell vernachlässigt wird. Daraus folgt:

$$a = F_{SE} + F_{SM}$$

Die angepasste Matlab-Funktion sieht wie folgt aus:

```
function a = Beschleunigung(xAktuell)

xE = xAktuell / norm(xAktuell);   % Einheiitsvektor fuer xAktuell
eSE = xE * (-1);          % Einheitsvektor vom Satelliten zu Erde - umgekehrte Richtung zu xE

xM = [0; -380000000];     % Mondposition
vSM = xM - xAktuell;      % Vektor vom Satelliten zum Mond
rSM = norm(vSM);
eSM = vSM / norm(vSM);    % Einheitsvektor vom Sat. zu Mond

r = norm(xAktuell);       % Abstand von Erdmitte zum Satellit

mE = 5.9736 * 10^24;      % Erdmasse in [kg]
mM = 7.3480 * 10^22;      % Mondmasse in [kg]
G = 66.743 * 10^-12;      % Gravitationskonstante in [m^3/kg*s^2]

F_SE = G * mE / r^2 * eSE; % Kraft auf Sat. von der Erde
F_SM = G * mM / rSM^2 * eSM; % Kraft auf Sat. vom Mond

a = F_SE + F_SM;          % Beschleunigung
```

Das Gesamtmodell kann man der folgenden Abbildung entnehmen:

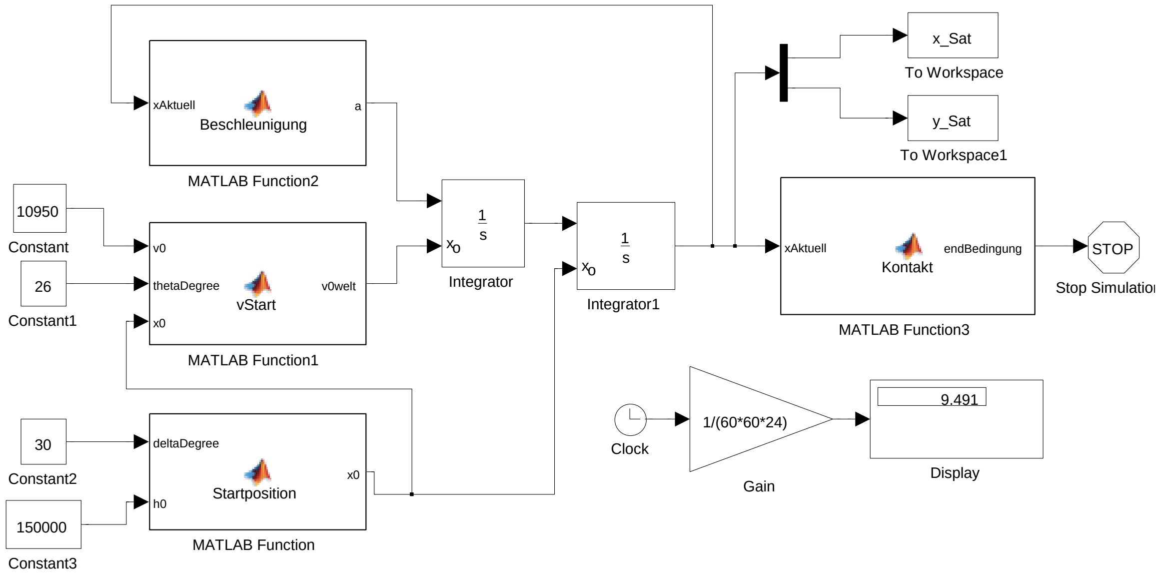

Abbildung 1.5: Gesamtmodell in Matlab/Simulink

Versuchsdurchführung

Für die Durchführung der Simulation wurden folgende Einstellungen verwendet:

$$\delta_0 = 30^\circ, h_0 = 150km$$

Bei einer Startgeschwindigkeit $v_0 = 10,95\frac{km}{s}$ und einer Neigung $\theta = 26^\circ$ fliegt der Satellit von der Erde um den Mond in einer 8-förmigen Schleife. Dabei dauert die Mission 9,491 Tage:

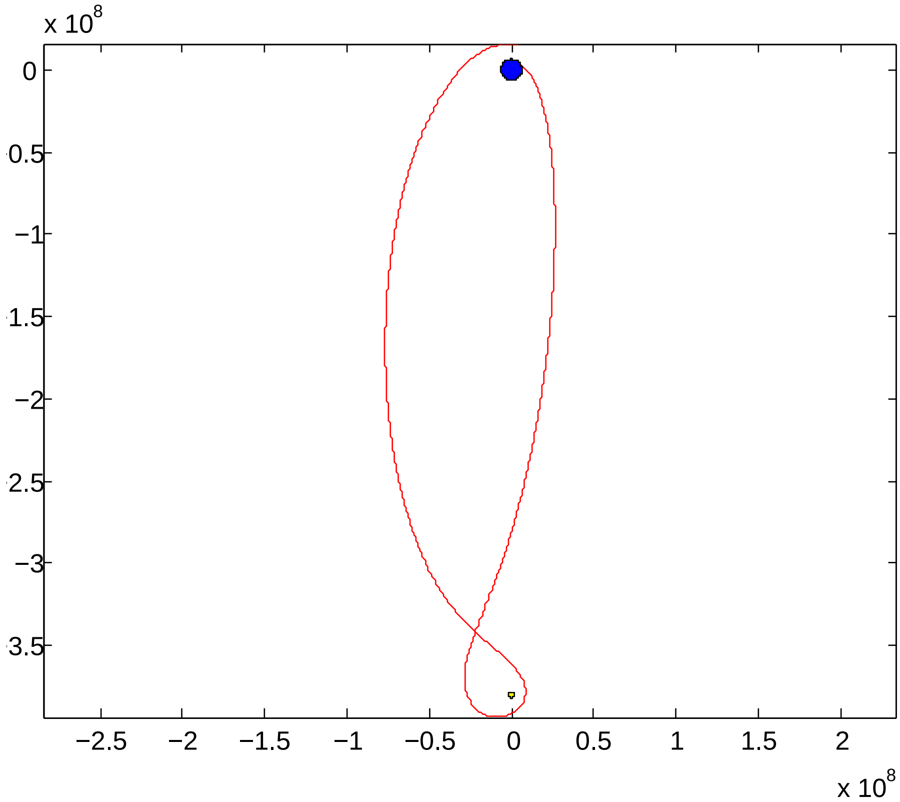

Abbildung 1.6: Mondmission