



## Projektaufgabe A Simulation

Fakultät Maschinenbau  
Simulation  
Fallender Ball

Semester:	<u>SS 2020</u>
Gruppe:	<u>9</u>
Datum:	<u>25.04.2020</u>

## Inhalt

Inhalt.....	2
1 Spezifikation und Vereinfachung.....	3
1.1 Spezifikationen Skript.....	3
1.2 Spezifikationen Simulation.....	3
1.3 Vereinfachung .....	3
2 Mathematische Gleichungen .....	4
2.1 Einheitenverzeichnis.....	4
2.2 Formelverzeichnis.....	4
2.3 Startparameter und Konstanten .....	4
2.4 Herleitung der Bewegungsgleichungen .....	5
3 Simulink Modell.....	6
4 Ergebnisse.....	7
5 Validierung .....	8

# 1 Spezifikation und Vereinfachung

## 1.1 Spezifikationen Skript

Die Skript Datei für Matlab soll sämtliche für die Simulation benötigten Konstanten und Startparameter bereitstellen. Für die Startparameter Geschwindigkeit  $v_0$  und den Wurfwinkel  $\alpha$  soll ein Interface für die Eingabe durch den Benutzer bereitgestellt werden.

Des Weiteren soll der Benutzer über den Status des Programmablaufs kontinuierlich über Konsolenausgaben informiert werden. Die numerischen Ergebnisse der Simulation werden gespeichert und über zwei Plots dargestellt. Diese beinhalten erstens: Weg und Geschwindigkeit in Y-Richtung über die Zeit, sowie zweitens: den Weg des Balls in Y Richtung über Weg in X Richtung.

Der Matlab Skript wird in der Version R2019b (oder älter) ausgeführt und ist mit ausreichend Kommentaren für die Nachvollziehbarkeit versehen.

## 1.2 Spezifikationen Simulation

Die Simulation muss für diese Anwendung sämtliche Konstanten und Startparameter derart zusammenführen und integrieren, dass eine lückenlose Darstellung der Bewegungsgleichungen und die Integrationen dieser Differentialgleichungen dargestellt werden können. Die so generierten zeitabhängigen Variablen werden als Wertearray an den Workspace des Matlab Skripts zurückgegeben, um die Berechnung der Ausgabeplots zu ermöglichen.

Hierzu gehört auch die Wahl eines geeigneten Zeitschrittwertes und das Umsetzen eines zentrischen Stoßes beim Aufprall des Balls auf das Nullniveau mit der Stoßzahl  $e$ .

Die Schaltungsblöcke und Signale sind hinreichend benannt.

## 1.3 Vereinfachung

Im Vergleich zur Realität wurden mehrere Vereinfachungen getroffen:

1. Der Luftwiderstand bzw. die Oberfläche des Balls werden vernachlässigt.
2. Der Stoß beim Aufprall erfolgt unelastisch; die Plastizität des Balls und Bodens werden nicht berücksichtigt, ebenso wird die Reibung zwischen beiden Körpern nicht berücksichtigt.
3. In der Realität hätte die Corioliskraft einen Einfluss auf die dreidimensionale Flugbahn des Balls, und würde so selbst in einer zweidimensionalen Darstellung einen Einfluss auf die Bewegung des Balls ausüben.
4. Innerhalb der Simulation wird die Bewegung des Balls Richtungsweise aufgeteilt. So werden Vektor- und Matrixoperationen vermieden.
5. In der Berechnung wird auf das Umwandeln des Winkels in rad verzichtet. Stattdessen wird die trigonometrischen Matlab Funktionen für Winkel in Grad verwendet.

## 2 Mathematische Gleichungen

### 2.1 Einheitenverzeichnis

Name	Formelzeichen	SI
Weg	$x, y, h$	m
Geschwindigkeit	$v$	$\text{m} \cdot \text{s}^{-1}$
Beschleunigung	$a$	$\text{m} \cdot \text{s}^{-2}$
Stoßzahl	$e$	Dimensionslos
Winkel	$\alpha$	$\text{Grad} = \frac{180^\circ}{\pi} \text{rad}$

### 2.2 Formelverzeichnis

Bezeichnung	Formel
Geschwindigkeit	$v = \frac{ds}{dt} = \dot{s}$
Beschleunigung	$a = \frac{dv}{dt} = \dot{v} = \ddot{s}$
Wurfhöhe	$h = \frac{v_0^2 \sin^2 \alpha}{2g}$
Wurfweite	$L = \frac{v_0^2 \sin^2 \alpha}{g}$
Stoß gegen starre Wand	$v_{\text{neu}} = -e * v$

### 2.3 Startparameter und Konstanten

Bezeichnung	Wert (min/max)	Einheit
Wurfwinkel (alpha)	Variabel (55°-65°)	$\text{Grad} = \frac{180^\circ}{\pi} \text{rad}$
Stoßzahl (e)	0,74	Dimensionslos
Starthöhe (h)	Variabel (22)	m
Startgeschwindigkeit ( $v_0$ )	Variabel (1-68)	$\text{m} \cdot \text{s}^{-1}$
Erdbeschleunigung (g)	9,81	$\text{m} \cdot \text{s}^{-2}$

## 2.4 Herleitung der Bewegungsgleichungen

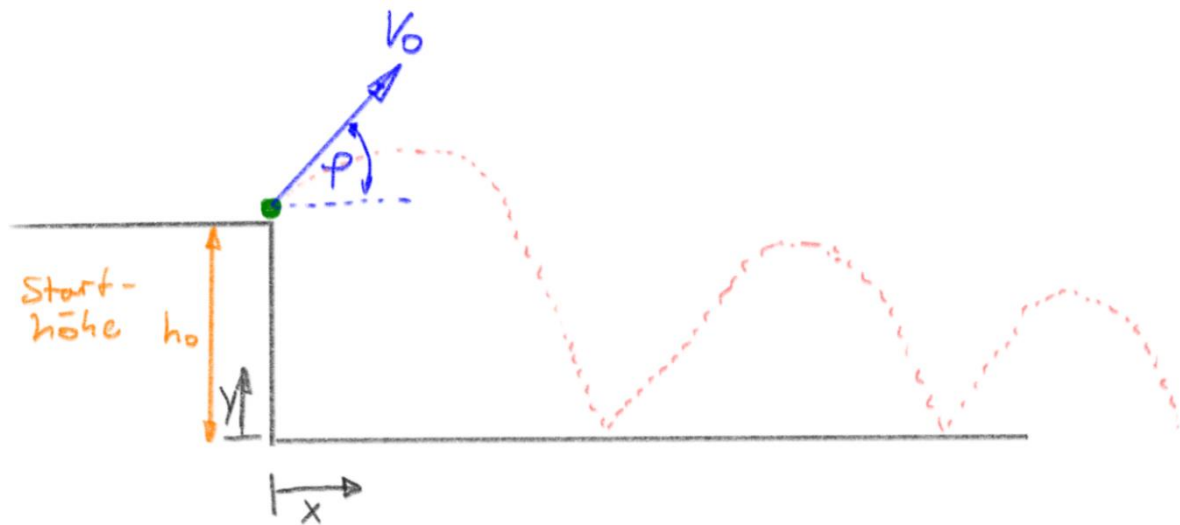


Abb. 1 – Skizze Aufgabenstellung

Freischneiden ergibt nach dem Ansatz von Newton mit den oben spezifizierten Vereinfachungen folgende Gleichung für die y-Richtung:

$$\begin{aligned}
 m * \ddot{y} &= -F_G = -m * g \\
 &\triangleq \ddot{y} = -g \\
 &\triangleq \dot{y} = -g * t + \dot{y}_0 \\
 &\triangleq y = -\frac{1}{2}g * t^2 + \dot{y}_0 * t + y_0
 \end{aligned}$$

Und für die x-Richtung:

$$\begin{aligned}
 m * \ddot{x} &= 0 \\
 &\triangleq \dot{x} = \dot{x}_0 \\
 &\triangleq x = \dot{x}_0 * t + x_0
 \end{aligned}$$

Die Startgeschwindigkeiten in x- und y-Richtungen ergeben sich aus der geometrischen Zerlegung der Startgeschwindigkeit in ihre vertikale und horizontale Komponente, die Starthöhe ist nach Aufgabenstellung konstant und die Wurfweite wird ab dem Abwurfpunkt gemessen, damit erhält man für die beiden Bewegungsgleichungen:

$$\begin{aligned}
 y &= -\frac{1}{2}g * t^2 + v_0 * \sin(\varphi) * t + 22 \text{ m} \\
 x &= v_0 * \cos(\varphi) * t
 \end{aligned}$$

### 3 Simulink Modell

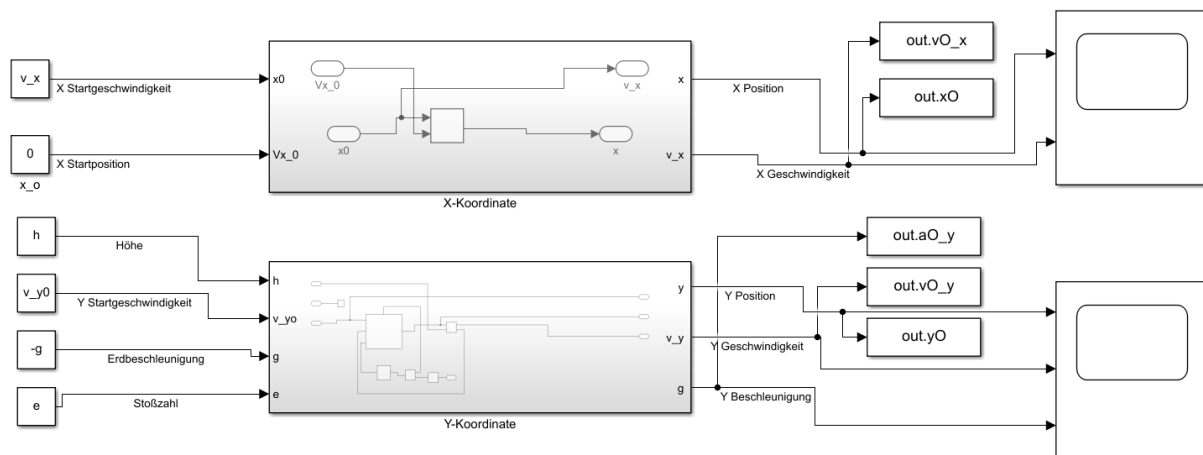


Abb. 2 - Simulink Modell

Um das Modell übersichtlicher zu gestalten wurde das Modell in die Subsysteme X-Koordinate und Y-Koordinate aufgeteilt. Das System X-Koordinate führt eine Integration der Geschwindigkeit des Balls in X-Richtung mit der Startgeschwindigkeit in X-Richtung als Konstante durch, um die zeitabhängige X-Position des Balls als Signal zu erhalten.

Etwas komplexer stellt sich das System Y-Koordinate dar: hier wird zunächst  $g$  mit Startgeschwindigkeit des Balls in Y-Richtung als Konstante in Geschwindigkeit in Y-Richtung integriert. Anzumerken ist hier ein Trigger für den Neustart des Signals, wenn die Y-Position 0 erreicht. Hierbei ist zu beachten, dass der letzte Wert für die Y-Geschwindigkeit zum Triggerzeitpunkt gespeichert, mit der Stoßzahl  $e$  verrechnet und als neuen Startwert in den Integrationsblock für  $g$  gegeben wird.

Aus dem so erhaltenen Signal „Geschwindigkeit in Y-Richtung“ wird über eine weitere Integration, welche nur beim ersten Durchlauf den Startparameter  $h$  erhält, das Signal für die Y-Position generiert. Zu beachten ist außerdem der zusätzliche Ausgang für den Trigger des Beschleunigungs-Integrationsblocks.

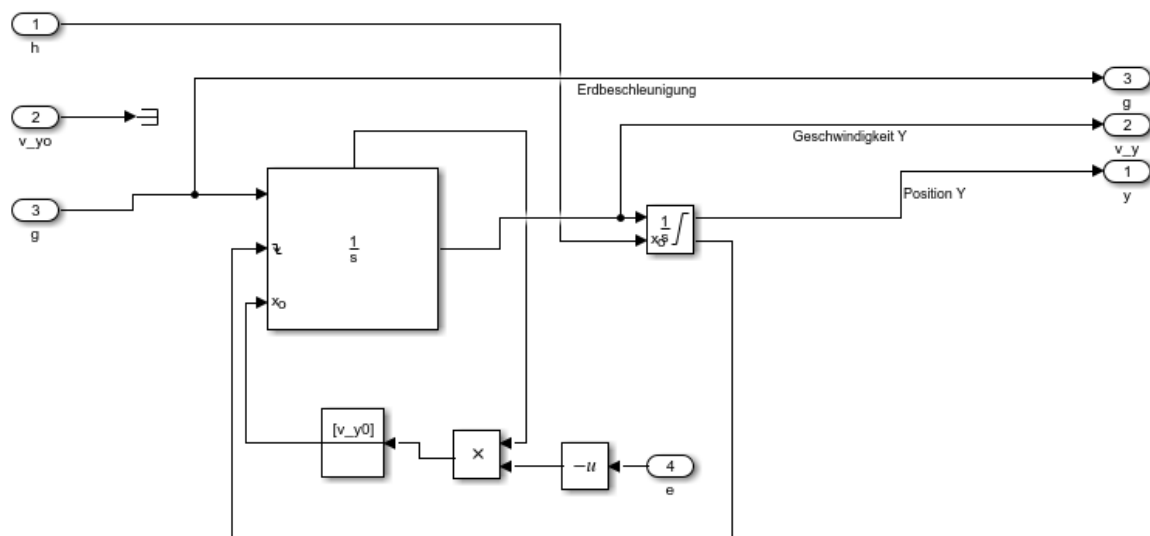


Abb. 3 - Subsystem Y-Koordinate

## 4 Ergebnisse

Die Ausführung des Simulink Modells erfolgt unter Verwendung einer Fixed-Step Size von  $10^{-4}$  s. In Abb. 4. sind die Eingaben für die Startparameter und die Statusausgaben des Programms zu erkennen.

```
Command Window

Saved alpha[°]: 60
Saved v_0[m/s]: 40
Inputs gültig!
Simulation startet...
Simulation abgeschlossen...
Plots werden erstellt...
Plots erstellt. Skript Beendet
fx >>
```

Abb. 4 - Ausgaben Programmstatus

Abbildung 5 zeigt den Plot der senkrechten Geschwindigkeit sowie der Höhe und der senkrechten Beschleunigung über der Zeit:

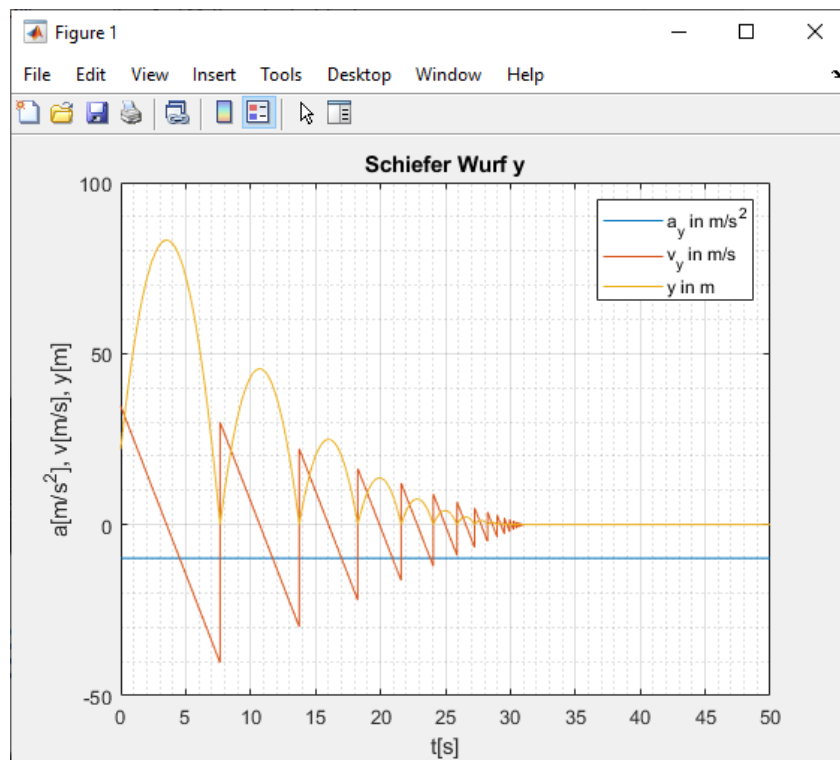


Abb. 5 - Figure 1

In Abbildung 6 ist der senkrechte Weg des Balls über dem horizontalen Weg dargestellt:

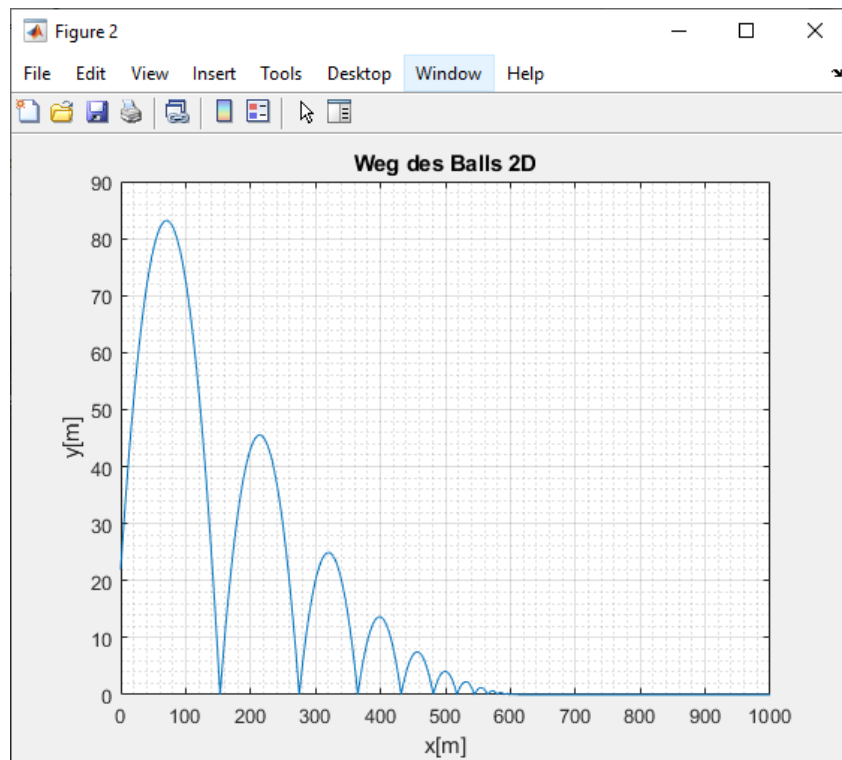


Abb. 6 - Figure 2

## 5 Validierung

Die Validierung des Simulink-Modells soll wie die Berechnung getrennt für die horizontale und die vertikale Bewegungsrichtung erfolgen. Für die senkrechte Bewegungsrichtung wird der Zeitpunkt des ersten Aufschlags auf dem Boden aus Figure 1 abgelesen sowie die neue vertikale Geschwindigkeit nach dem hier erfolgenden Stoß abgelesen:

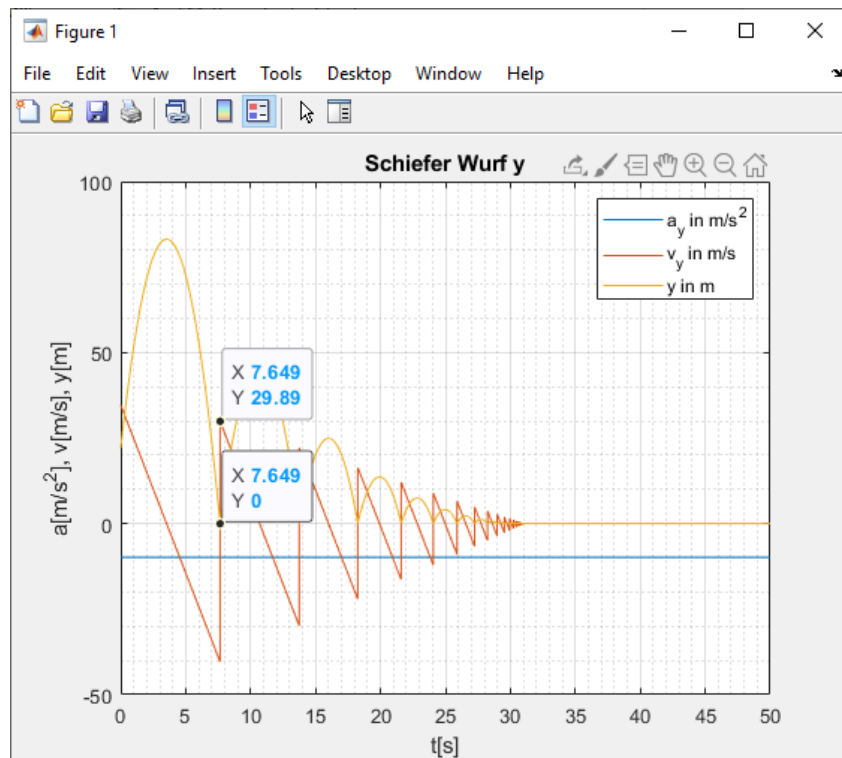


Abb. 7 - Figure 1, Validierung



In der Formel für die vertikale Bewegung aus Abschnitt 2.4 wird die Höhe gleich 0 gesetzt und die Startgeschwindigkeit und –winkel aus Abschnitt 4 eingesetzt, damit erhält man:

$$\begin{aligned} 0 \text{ m} &= -\frac{1}{2}g * t^2 + 40 \frac{\text{m}}{\text{s}} * \sin(60^\circ) * t + 22 \text{ m} \\ &\triangleq 4,905 \frac{\text{m}}{\text{s}^2} * t^2 - 34,64 \frac{\text{m}}{\text{s}} * t - 22 \text{ m} = 0 \\ &\triangleq t^2 - 7,06 \text{ s} * t - 4,49 \text{ s}^2 = 0 \end{aligned}$$

Damit erhält man aus der pq-Formel:

$$\begin{aligned} t_{1|2} &= \frac{7,06}{2} \text{ s} \pm \sqrt{\left(-\frac{7,06}{2} \text{ s}\right)^2 + 4,49 \text{ s}^2} \\ &\triangleq t_{1|2} = 3,53 \text{ s} \pm \sqrt{16,95 \text{ s}^2} \\ &\triangleq t_{1|2} = 3,53 \text{ s} \pm 4,12 \text{ s} \\ &\Rightarrow t_1 = -0,59 \text{ s} \quad \cancel{\text{↯}} \\ &\Rightarrow t_2 = 7,65 \text{ s} \end{aligned}$$

7,65 Sekunden entsprechen mit hinreichender Genauigkeit dem Ergebnis von 7,649 Sekunden aus der Simulation.

Die senkrechte Geschwindigkeit beträgt demnach zum Zeitpunkt des Aufschlags:

$$\begin{aligned} \dot{y} &= -g * 7,65 \text{ s} + 34,64 \frac{\text{m}}{\text{s}} \\ &\triangleq \dot{y} = -9,81 \frac{\text{m}}{\text{s}^2} * 7,65 \text{ s} + 34,64 \frac{\text{m}}{\text{s}} = -40,41 \frac{\text{m}}{\text{s}} \end{aligned}$$

Und nach dem Stoß gegen die starre Wand (Boden):

$$\begin{aligned} \dot{y}_{\text{neu}} &= -e * \dot{y} \\ &\triangleq \dot{y}_{\text{neu}} = -0,74 * -40,41 \frac{\text{m}}{\text{s}} = 29,90 \frac{\text{m}}{\text{s}} \end{aligned}$$

Dies deckt sich ebenfalls mit dem aus Abb. 5 abgelesenen Wert, somit ist das Modell für die senkrechte Bewegung validiert.

Für die Validierung der horizontalen Bewegung wird ebenfalls der Zeitpunkt des ersten Aufschlags auf dem Boden verwendet, das Ergebnis der Simulation ist in Abbildung 8 abzulesen:

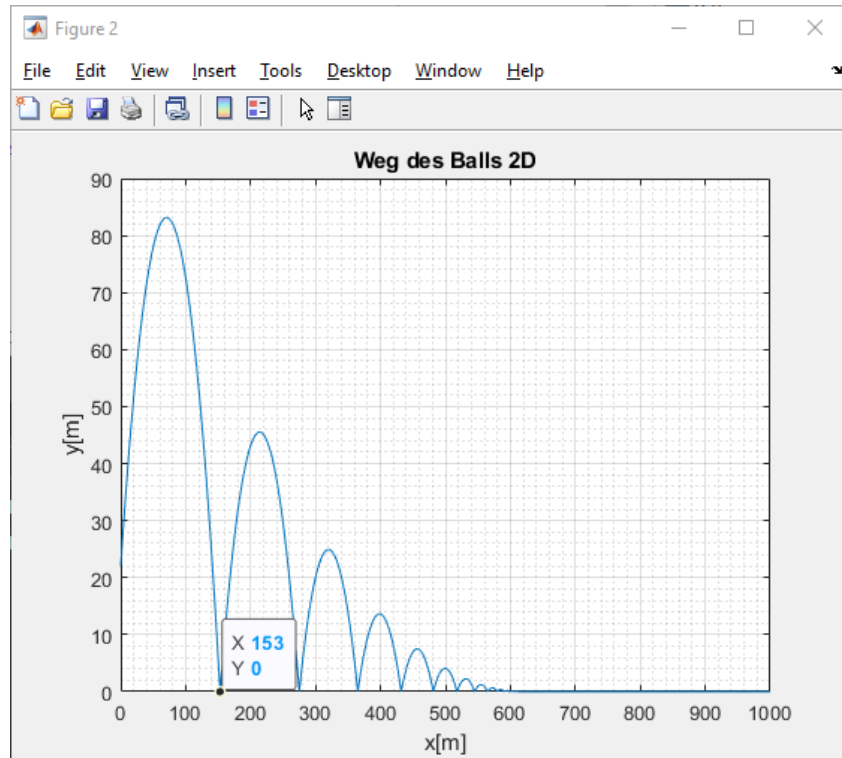


Abb. 8 - Figure 2, Validierung

Auch hier werden die Startparameter aus Abschnitt 4 in die Bewegungsgleichung aus Abschnitt 2.4 eingesetzt:

$$x = 40 \frac{m}{s} * \cos(60^\circ) * 7,65 s = 153 m$$

Somit ist auch das Modell für die horizontale Bewegung validiert.