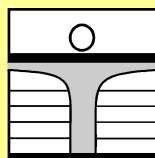


TU LIBEREC
Hálkova 6
461 17 Liberec 1, CZ



**HOCHSCHULE
ZITTAU/GÖRLITZ**
(FH) – University of Applied Science
Theodor-Körner-Allee 16, D - 02763 Zittau

Fakulta mechatroniky a
mezioborových inženýrských studií

Fachbereich Elektro – und
Informationstechnik

Das internationale Projekt
„ENTWURF UND PROJEKTIERUNG MECHATRONISCHER SYSTEME“

Einführung in Simulink

Arbeitsversion

Osvald Modrlák
Frank Worlitz

April, 2006



Katedra řídicí techniky



1.	Vorwort	2
2.	Matlab's Anwenderumgebung (Environment)	3
3.	Benutzeroberfläche von MATLAB	4
4.	Simulink	8
5.	Bildung eines einfachen Simulinkprogramms	13
5.1.	Mathematisches Modell eines Federsitzes	13
5.2.	Wirkung einer Kraft	13
5.2.1.	Aufstellung eines SIMULINK Programms mit Übertragungsfunktion ...	13
5.2.2.	Aufstellung eines SIMULINK Programms, wenn die Beschreibung durch Differentialgleichung gegeben wird	17
5.3.	Wirkung des Fahrers	19
5.4.	Wirkung und Modellierung der Koeffizient $c = c(y')$	21
6.	Blockparametrisierung	23
6.1.	Parametrierung der SIMULINK- Simulation	23
6.2.	Parametrierung des Blocks „Constant“	26
6.3.	Parametrierung des Blocks „Display“	26
6.4.	Parametrierung des Blocks Slidergains	27
6.5.	Parametrierung des Blocks Step	27
6.6.	Parametrierung des Blocks „Scope“	28
6.7.	Parametrierung des Blocks „To File“	28
6.8.	Parametrierung des Blocks „Pid Controller“	29
6.9.	Parametrierung des Blocks „Pid Controller With Approximate Derivation“	30
6.10.	Parametrierung des Blocks „Transfer Fcn“	30
6.11.	Parametrierung des Blocks „Repeating Sequence“	31
6.12.	Parametrierung des Blocks „Product“	31
6.13.	Parametrierung des Blocks „Fcn“	32
6.14.	Parametrierung des Blocks „MATLAB Fcn“	33
7.	Untersysteme	34
8.	Untersuchung der realen Objekte	36
9.	Literatur	38

Der vorgelegte Studententext ist eine Einführung in die SIMULINK- Programmierung, wo ein Programm mit Hilfe der Blocks und deren Verbindung eine Lösung von Differentialgleichungen bestimmt wird. Durch Parametrisierung werden den Blöcken entsprechende Werte des Modells zugeordnet. Die Auswahl des Inhalts hängt mit der Aktivitäten der Studenten bei Austauschpraktika zusammen. Dieser Text stellt selbstverständlich keinen erschöpfenden Überblick über das SIMULINK Programmierung. Er soll eher einen vereinfachten Einblick in die ganze Problematik anbieten.

Einführung in SIMULINK und MATLAB

1. VORWORT

MATLAB stellt eine wissenschaftlich-technische Software für die Simulation von dynamischen Prozessen dar. Grundsätzlich kann man sagen, dass **MATLAB** enthält:

- Bibliothek mit einer Vielzahl von mathematischen Funktionen
- Bausteine für Lösung der wissenschaftlich-technischen Aufgaben
- Entwicklungsmittel für Algorithmusbildung
- Modellbildung und Simulation
- Graphik 2D und 3D
- Datenanalyse und Datenauswertung
- Interaktive Arbeitsweise mit Helffunktionen

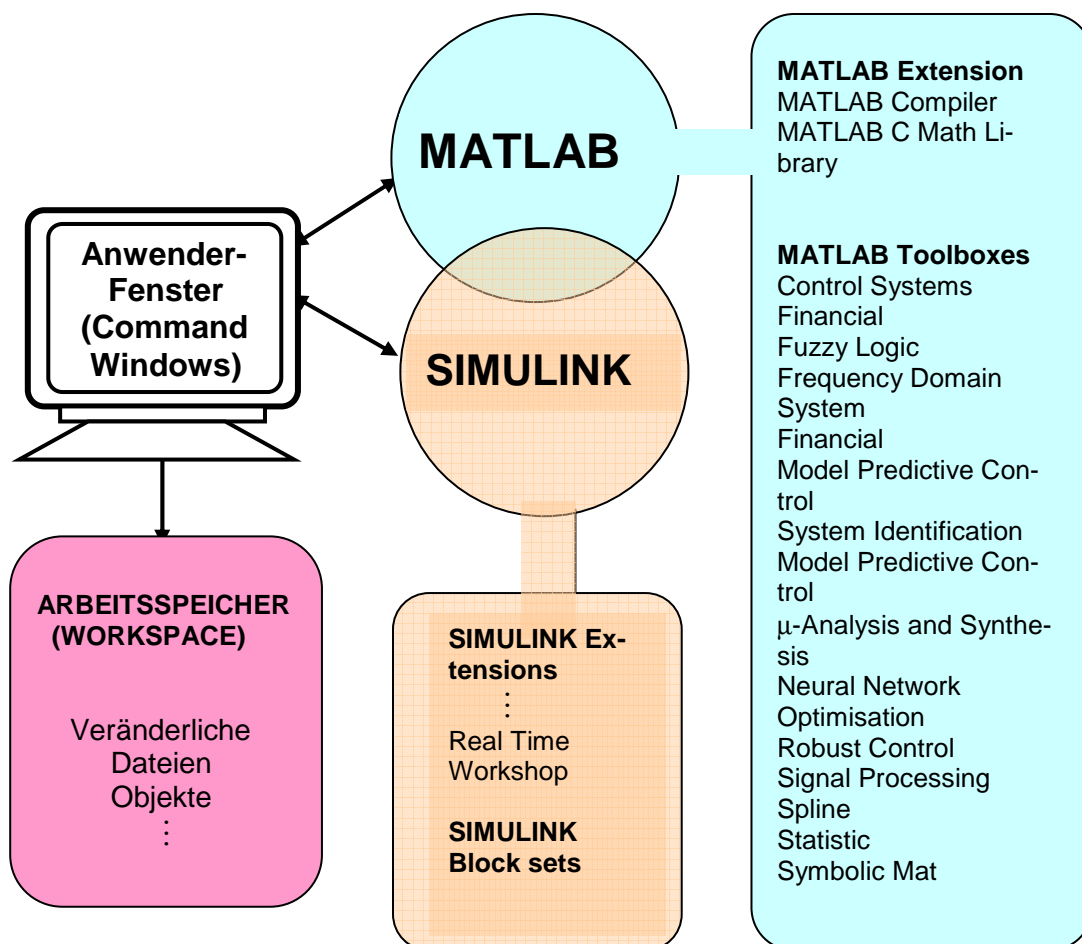


Bild 1. MATLAB/SIMULINK Tools und Erweiterungen

Ein Teil dieses Produkts ist graphisch-orientierende Software **SIMULINK**, die mit Hilfe der Funktionsblöcke das zu simulierende System graphisch nachbilden. Nach der Parametrisierung der Blöcke kann die Simulation gestartet werden.

2. MATLAB'S ANWENDERUMGEBUNG (ENVIRONMENT)

Auf dem Bildschirm von MATLAB arbeitet der Anwender in dem **Anwenderfenster** (Command Windows).

Der MATLAB bildet bei jeder Sitzung (MATLAB Session) einen **Arbeitsspeicher** (Workspace), wo Veränderliche, Dateien, Objekte usw. binnen der Sitzung gespeichert sind.

Nach Beendigung jeder Sitzung wird der Arbeitsspeicher automatisch gelöscht. Das bedeutet, dass es vor dem „Exit“ nötig ist, die Ergebnisse, Dateien oder Graphische Darstellung zu speichern.

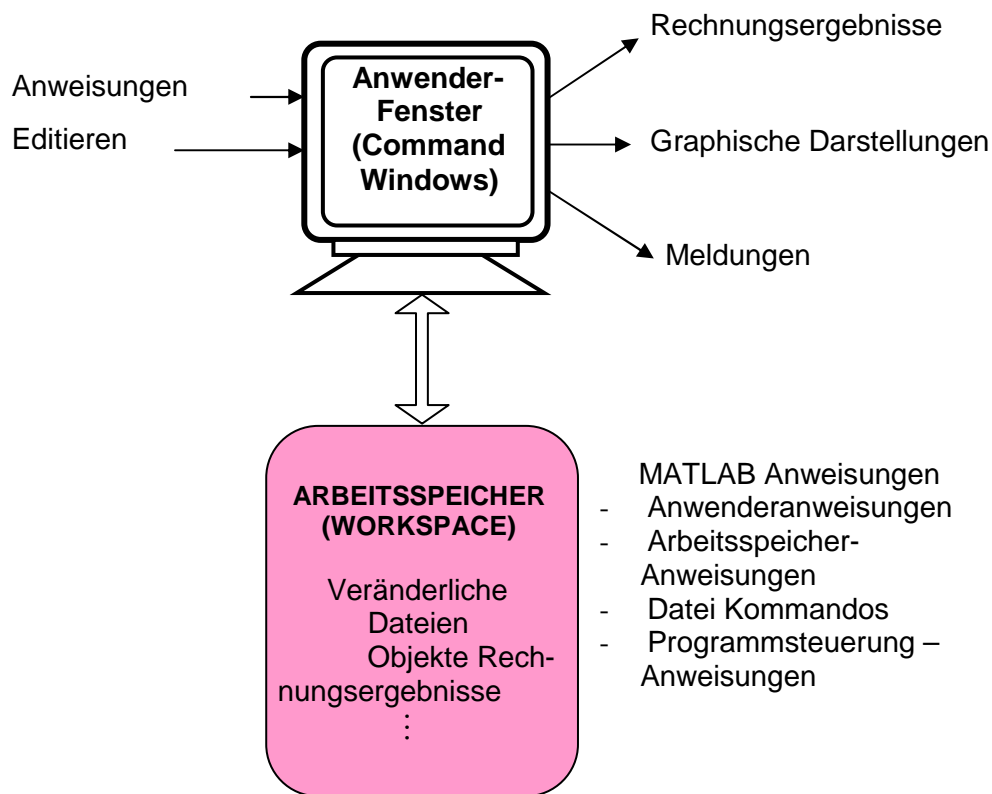


Bild 2. MATLAB'S Struktur von Anwenderfenster und Arbeitsspeicher

3. BENUTZEROBERFLÄCHE VON MATLAB

MATLAB – Programme werden in einer interaktiven Umgebung entworfen und geschrieben. Bei MATLAB- Start, der entweder über Doppelklick auf die MATLAB I-kone oder durch Eingabe des Befehls `matlab` im Kommando-Interpreter des Betriebssystems durchgeführt wird, wird seine Benutzeroberfläche folgend dargestellt, siehe bild 3.1. Die Struktur der Benutzeroberfläche entspricht dem Betriebssystem „Windows“. Man sieht Menü- und Symbolleisten, Fenster und Startknopf.

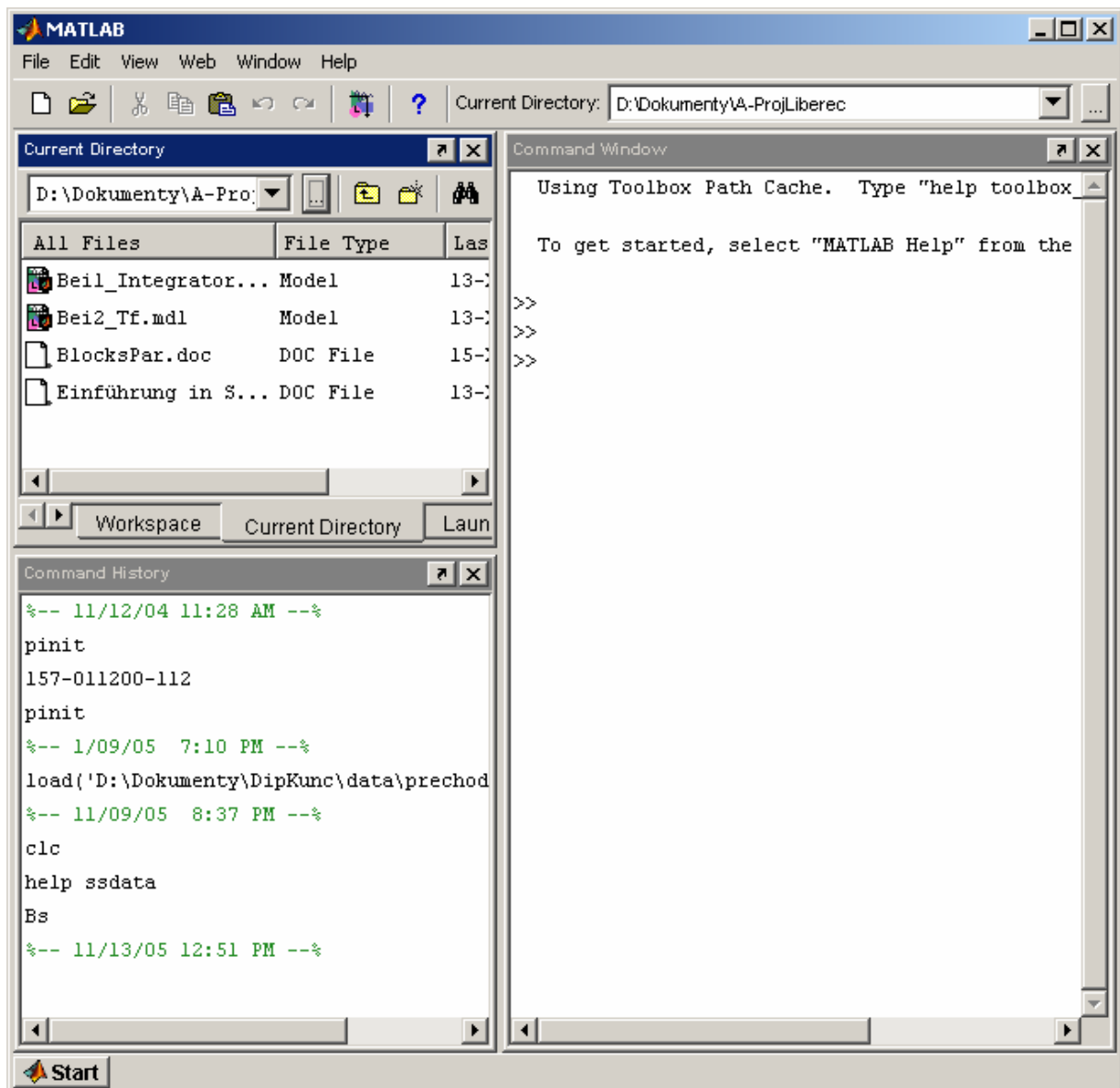






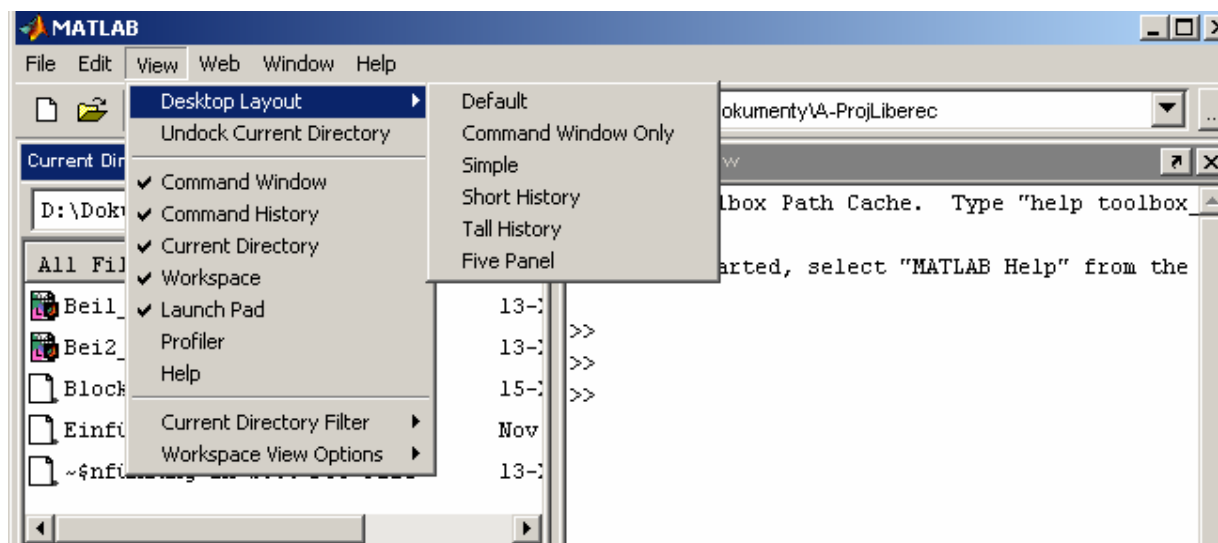
Bild 3.1 MATLAB Benutzeroberfläche

Die Benutzeroberfläche von MATLAB besteht aus:

- 1) einer Menüleiste,
- 2) einem Fenster mit den Registern:
 - a) Workspace
 - b) Current Directory

- c) Launch Pad
- 3) einem Kommandofenster,
- 4) einem Fenster „Command History“,
- 5) einer Symbolleiste, die enthält Symbole für:
 - a) einen Texteditor ,
 - b) ein "Open File" Symbol, ,
 - c) den "Simulink" Block Set Erweiterung, ,
 - d) help,
 - e) eine Editierung. 

1) Durch Roll-Menü „View“ in der Menüleiste wird die Benutzeroberfläche definiert.



3.2 Desktop Layout

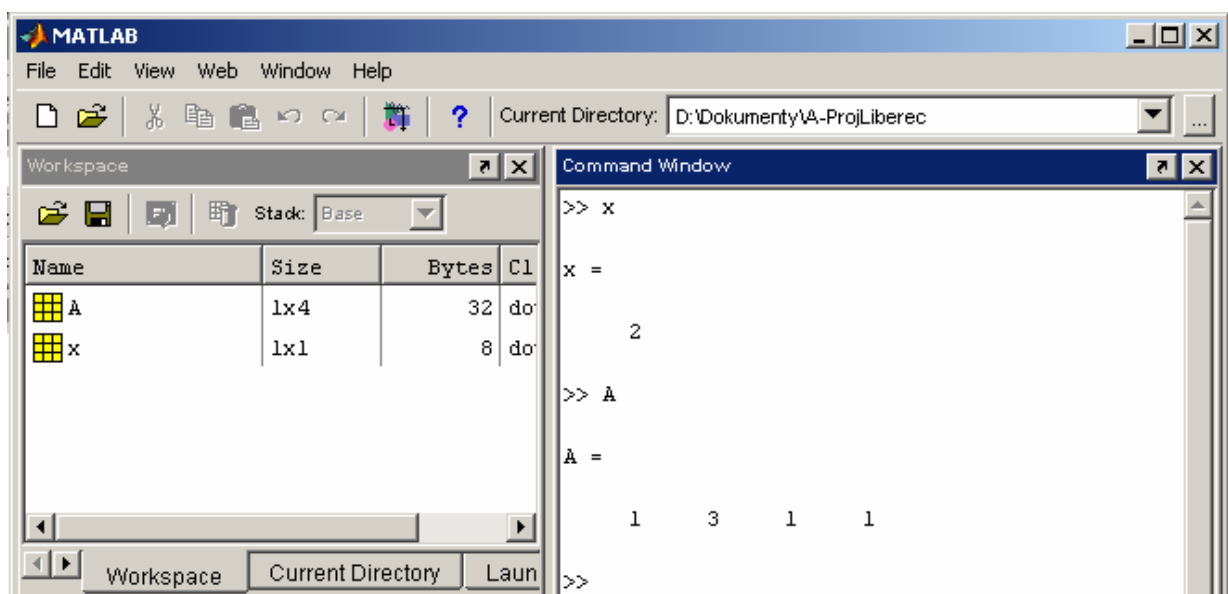


Bild 3.3 Auszug aus dem Workspace Register

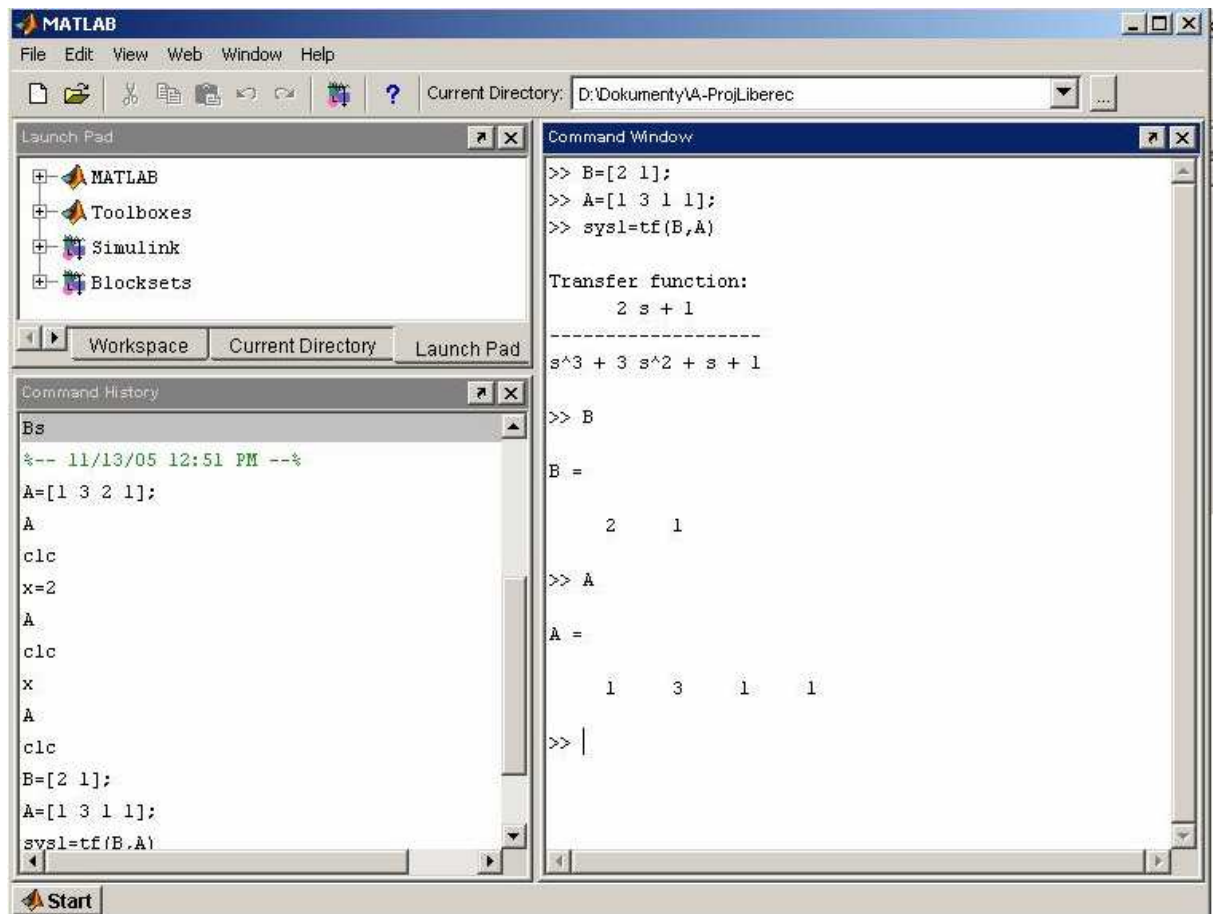
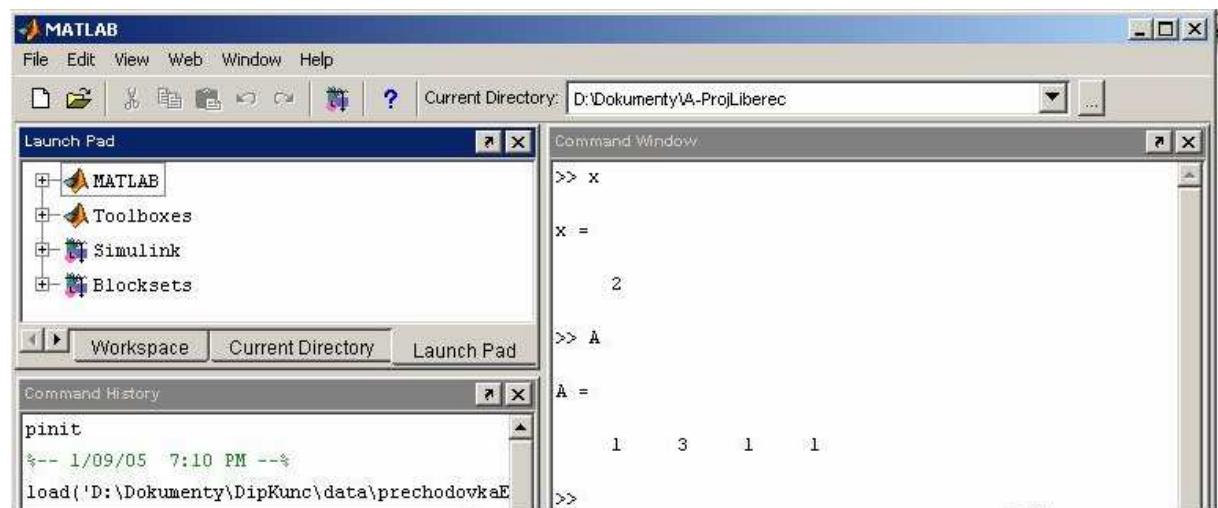


Bild 3.4 Kommandofenster

- 2) Registern:
 - a) Der „Workspace(Arbeitsspeicher)“ – ermöglicht das Anzeigen und Verändern von Variable
 - b) Die „Current Directory“ zeigt den Inhalt des aktuellen Verzeichnisses an.
 - c) Das „Launch Pad“ sowie der MATLAB-Sartknopf ermöglichen den Zugang zu allen installierten MATLAB-Produkten
- 3) Das Kommandofenster dient den interaktiven Eingabe von MATLAB- Befehlen.
- 4) Die „Command History“ Fenster enthält die Liste aller bisher ausgeführten Befehle.



3.5 Launch Pad



```

Command History
Aq.variable='q'
B=pol2mat(Aq)
qM
pol2mat(qM)
clc
%-- 11/09/05 1:20 PM --%
clc
help gensyn
clc
help gensyn
help pformat
help pol2mat
clc
%-- 11/14/05 12:28 PM --%

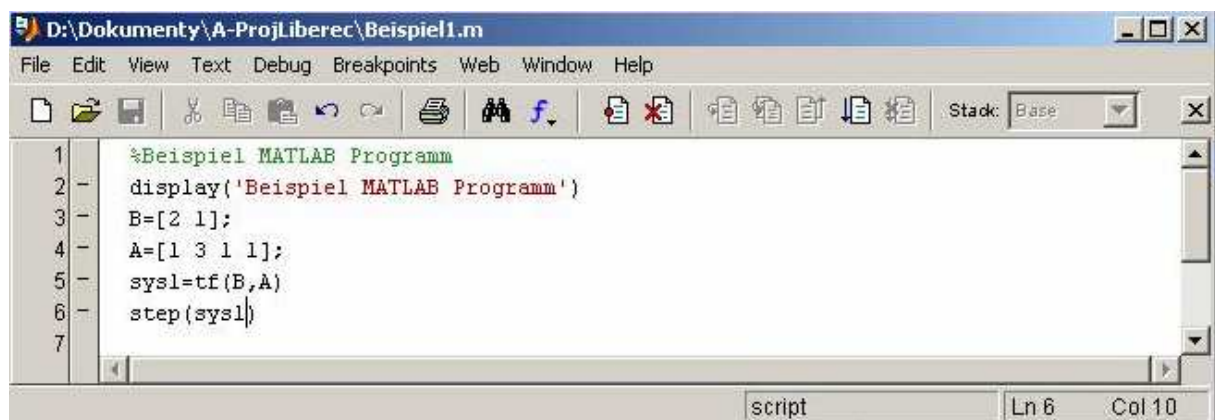
```

3.6 Command History

5) Die Symbolleiste

a) Texteditor

Mit diesem Texteditor kann man MATLAB –Funktionen und –Skripts erstellen.



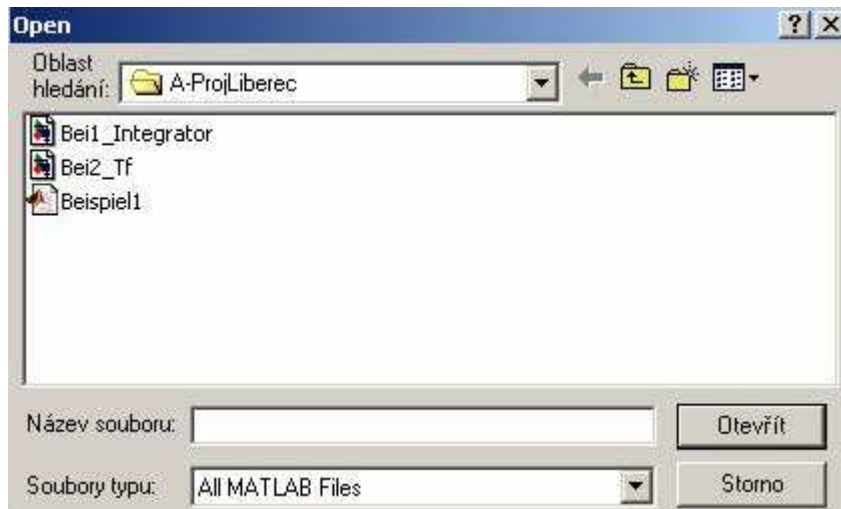
```

D:\Dokumenty\A-ProjLibrec\Beispiel1.m
File Edit View Text Debug Breakpoints Web Window Help
%Beispiel MATLAB Programm
1 display('Beispiel MATLAB Programm')
2 B=[2 1];
3 A=[1 3 1 1];
4 sys1=tf(B,A)
5 step(sys1)
6
7
script Ln 6 Col 10

```

3.7 Fenster Text Editor

b) ein "Open File" Symbol





3.8 Fenster „Open“

c) Help
d) eine Editierung



4. SIMULINK

SIMULINK ist eine Block-Set Erweiterung von MATLAB, die für Simulation von allgemeinen Prozessen bestimmt ist. Das Simulationsschema wird graphisch durch Funktionsblöcke gebildet, die vor der Simulation eingestellt werden müssen.

Durch die Anweisung `simulink` oder durch Doppelklicken auf das Symbol  öffnet sich das Fenster "Simulink Library Browser", das auf dem Bild 4.1 dargestellt ist. Das Fenster enthält „Simulink - Ikone“ mit seiner Blockbibliotheken und eine Reihe von Tools, die  mit der Ikone bezeichnet sind.

Durch Doppelklicken auf das Symbol einer Bibliothek öffnet sich ein neues Fenster,

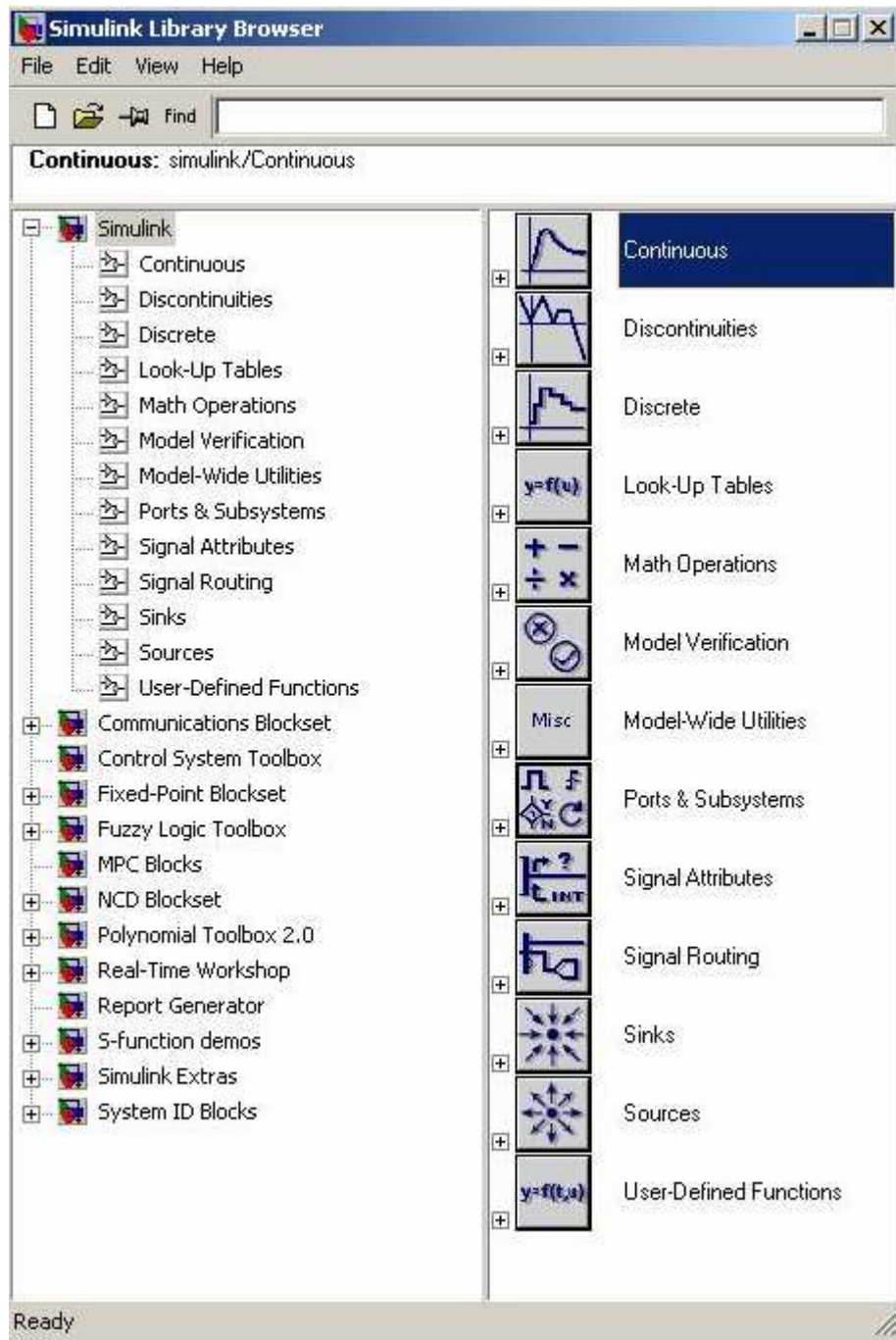


Bild 4.1 Simulink Fenster

das die Funktionsblöcke dieser Bibliothek enthält. Jeder Funktionsblock enthält eine englische Anweisung zur Konfiguration und einen englischen „**help**“. Wir werden nur die wichtigsten Bibliotheken beschreiben. Die Anwendungen der Blöcke und die Programmierung wird auf einem Beispiel vorgeführt (Kapitel 5).

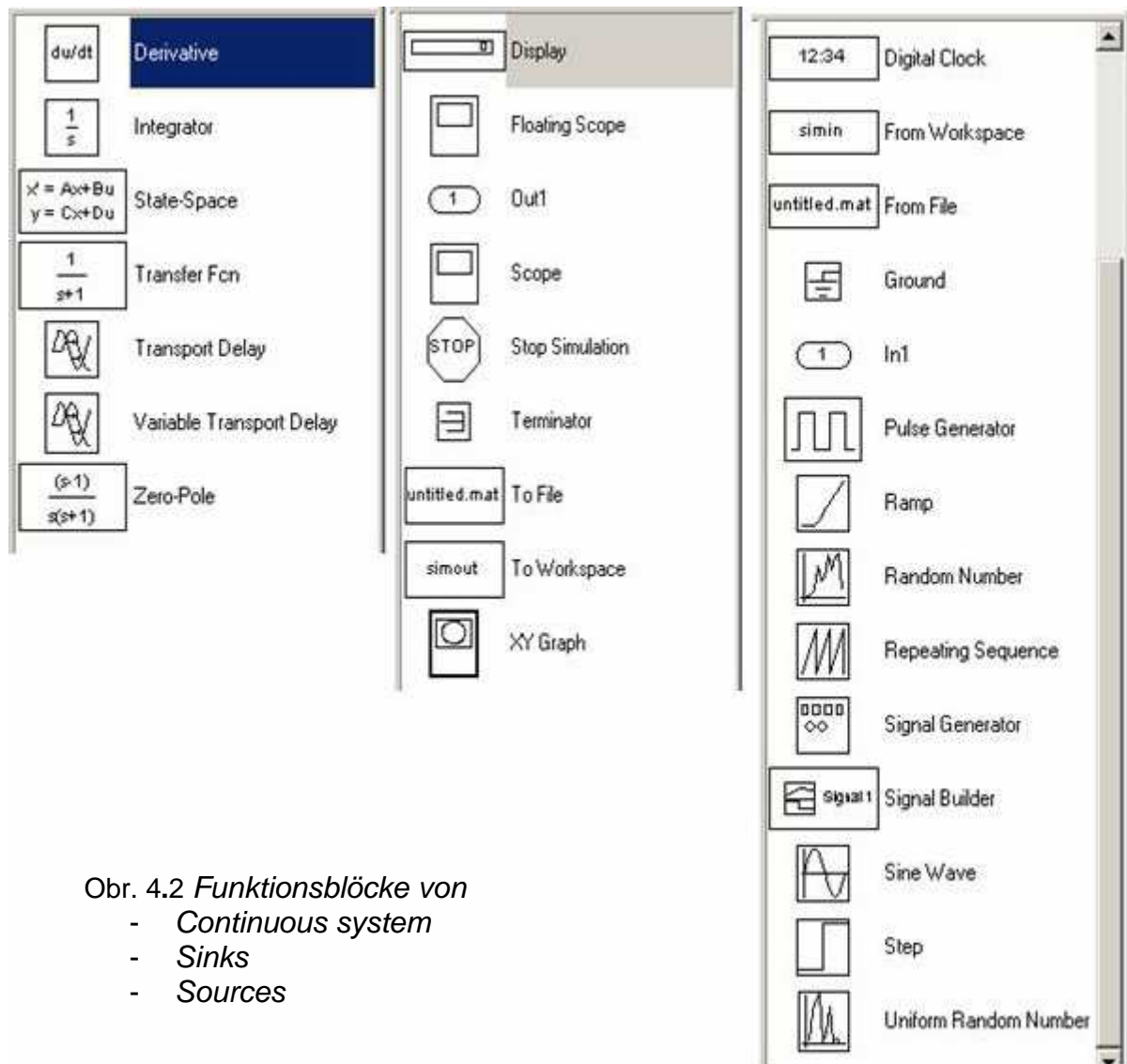
Auf dem Bild 4.2 sind die Bibliotheken „Continuous, Sinks und Sources“ geöffnet. Die Bibliothek „**Continuous**“ enthält die Integrations- und Derivationsblöcke. Weiter sind die Blöcke „State Space, Transfer Fcn, und Zero-Pole“ zur Verfügung gestellt, die lineare time-

invariante Systeme darstellen und Simulieren. Als Bestandteil dieser Bibliothek sind Blöcke „Transport Delay, Variable Transport Delay“, die Zeitverspätung modellieren können.

Continuous:

Sinks:

Sources:



Obr. 4.2 Funktionsblöcke von

- Continuous system
- Sinks
- Sources

Die Ergebnisse an verschiedenen Punkten des Simulationsschemas können entweder **graphisch oder alphanumerisch** dargestellt werden. Dazu werden die Blöcke aus der Bibliothek „**Sinks – Blöcke für Datendarstellung und Verarbeitung**“ benutzt.

Eine Graphische Darstellung der bearbeiteten Datei werden durch die Blöcke „Scope, Floating Scope, XY Graph“ gesichert. Alphanumerische Darstellung der Datei auf dem Bildschirm wird durch den Block „Display“ durchgeführt. Durch den Block „To File“ werden die Ergebnisse als eine Datei gespeichert. Mit Hilfe des Blocks „To Workspace“ werden Ergebnisse als eine Datei in dem Arbeitsspeicher gespeichert. Ein spezielles Signal ist der Block „Out1“, der ein Output-Port für ein Subsystem oder Model sichert.

Dynamische Systeme können verschiedene Eingangssignale als Erregung haben. Dazu dient die Bibliothek „**Source – Signalquelle**“. Als deterministischen Signalgeneratoren werden „Digital Clock, Pulse Generator, Ramp, Repeating Sequence, Signal Generator, Sine Wave, und Step“ benutzt. Als Zufalgeneratoren stehen Ihnen zu Verfügung die Blöcke „Random Number, Uniform Random Number“. Als Eingangssignale können auch verschiede-

den Arten von Dateien benutzt werden. Dazu sind die Blocks „From Workspace, From File“ geeignet. Spezialer Eingangssignal ist der Block „In1“, der die Verbindung von einem Subsystem sichert.

Auf dem Bild 4.3 ist ein Überblick über Blöcke der Bibliotheken „Signal Routing und Math Operations“ zu sehen. Bei der Darstellung der Signale ist es oft nötig Signale zu verbinden oder zu entbinden usw. Dazu dienen Blöcke der Bibliothek „**Signal Routing**“.

Signal Routing:

Math Operations

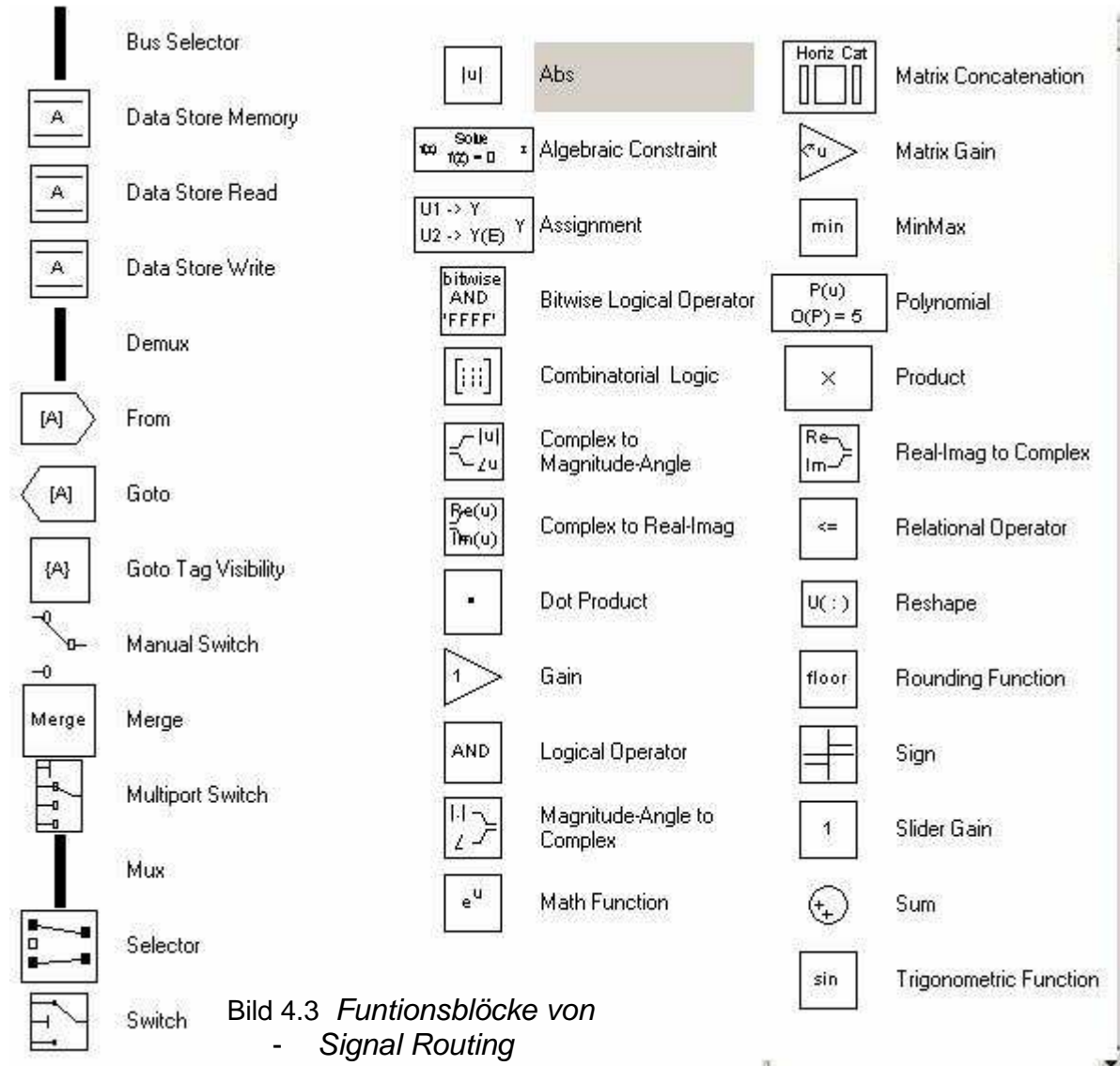
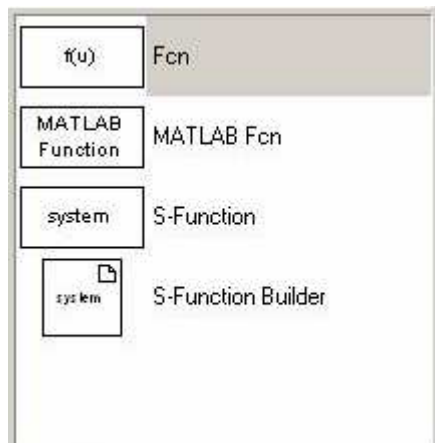


Bild 4.3 Funktionsblöcke von
- Signal Routing
- Math Operation

Blöcke für mathematische Operationen finden Sie in der Bibliothek „**Math Operations**“. Die Bibliothek „**User- Defined Function**“ ermöglicht den Anwender Blöcke zu benutzen, die eine von dem Anwender definierte Funktion, MATLAB Funktion oder andere definiert wird.

User -Defined Function

5. BILDUNG EINES EINFACHEN SIMULINKPROGRAMMS

Wir werden an einem einfachen Beispiel den Aufbau eines SIMULINK Programms demonstrieren.

5.1. MATHEMATISCHES MODELL EINES FEDERSITZES

Als dynamisches System wird ein Federsitz im Bild 5.1 betrachtet, der durch eine Sitzmasse m , Federstarrheit k und Dämpfung c charakterisiert wird. Das Ziel der Simulation ist, die vertikale Bewegung der Federsitze zu analysieren unter Wirkung:

- 1) einer Kraft F ,
- 2) des Fahrers, der sich auf den Sitz hinsetzt und der eine Fahrermasse mf besitzt.

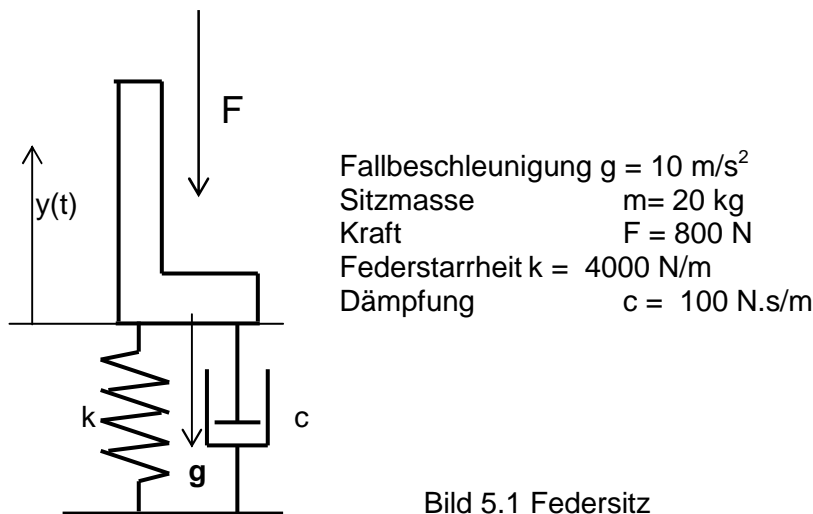


Bild 5.1 Federsitz

Das dynamische Verhalten von einem System kann allgemein durch differentiale Gleichungen beschrieben werden. Ein Federsitz nach dem Bild 5.1 kann man als Feder-Masse-System betrachten unter Wirkung der Kraft $F(t)$ und des Fallbeschleunigung g , der mit folgender Differentialgleichung beschrieben werden kann.

$$m \cdot \ddot{y}(t) = -c \cdot \dot{y}(t) - k \cdot y(t) - m \cdot g - F(t) \quad (5-1)$$

5.2. WIRKUNG EINER KRAFT

Die Gleichung 5.1 ist eine Beschreibung eines linearen zeitinvarianten Systems. Das Simulationsprogramm kann für ein mathematisches Modell in Form einer Übertragungsfunktion oder direkt der Differentialgleichung aufgestellt werden.

5.2.1. AUFSTELLUNG EINES SIMULINK PROGRAMMS MIT ÜBERTRAGUNGSFUNKTION

Lineare zeitinvariante Systeme können im Bildbereich durch Übertragungsfunktion beschrieben. Die Gleichung (5.1) kann umgeschrieben werden in die Form

$$\ddot{y}(t) + c/m \cdot \dot{y}(t) + k/m \cdot y(t) = -u(t), \quad (5-2)$$

wo $u(t) = g + F(t)/m$ ein Eingangssignal ist.

Bei Anfangsbedingung $y(0)' = y(0) = 0$ bekommen wir mit Hilfe der Laplace - Transformation die Übertragungsfunktion

$$Y(s)s^2 + Y(s)s \cdot c/m + Y(s) \cdot k/m = -U(s) \rightarrow G(s) = \frac{Y(s)}{U(s)} = \frac{-1}{s^2 + s \cdot (c/m) + k/m} \quad (5-3)$$

$$Y(s) = G(s) \cdot U(s) = \frac{-1}{s^2 + s \cdot (c/m) + k/m} \cdot U(s), \quad u(t) = g + 1/m \cdot F(t).$$

Wenn wir ein SIMULINK Programm für die Übertragungsfunktion (5.3) aufbauen wollen, werden wir folgende Blöcke brauchen:

- Transfer Fcn ...für die Übertragungsfunktion,
- Sum ...für die Summation von Signalen g und $F(t)/m$,
- Gain ...für die Verstärkung $1/m$,
- Step ...für das Signal $F(t)$,
- Konstant ...für die Fallbeschleunigung g ,
- Scope ...für die graphische Darstellung des Ausgangssignal $y(t)$.

Diese Blöcke sind in das Fenster „untitled“ zu ziehen.

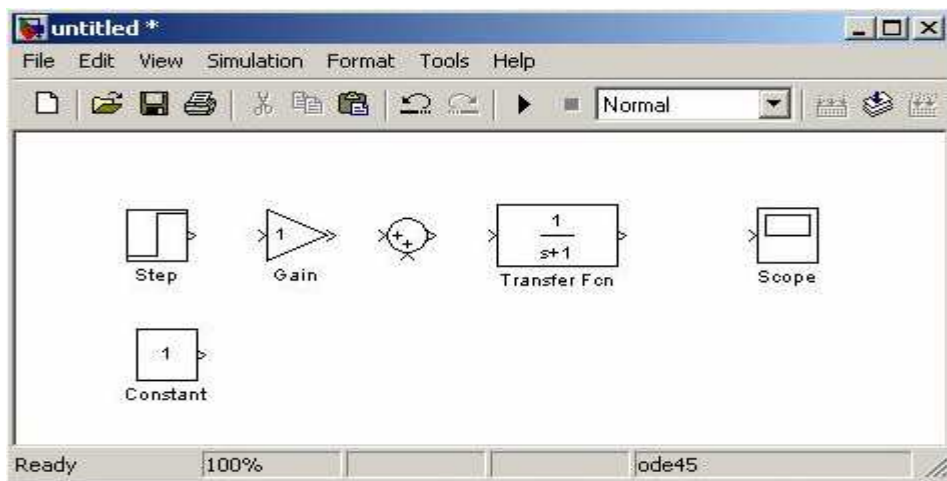


Bild 5.2. Die notwendigen Blöcke

Ein SIMULINK Programm wird durch die Verbindung der Blöcke gebildet werden, wie das auf dem Bild 5.3 dargestellt ist.

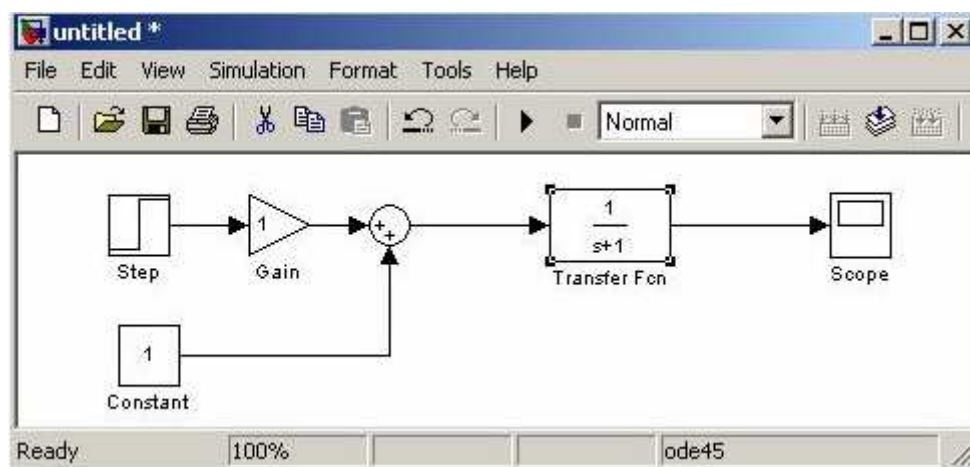
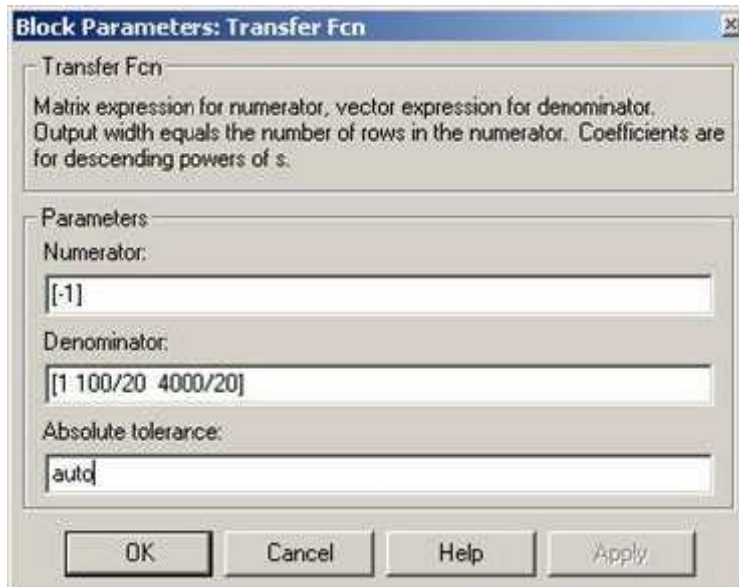


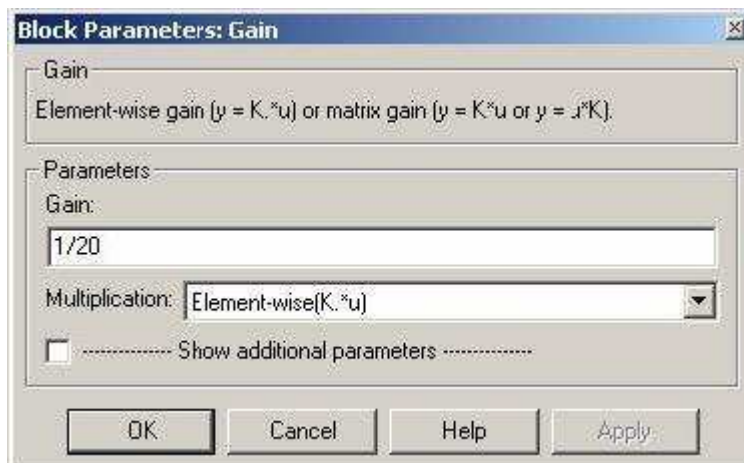
Bild 5.3 SIMULINK Programm ohne Parametrisierung

Danach müssen die Blöcke parametrisiert werden und das Programm unter einem Name gespeichert werden. Mit einem Doppelklick werden die Blöcke für die Parametrisierung geöffnet und entsprechende Einstellung kann vorgenommen werden. Um man sich eine Vorestellung über Parametrisierung der Blöcke machen könnte, ist diese für unser Programm Bild 5.4a Parametrisierung des Blocks „Transfer Fcn“



$$G(s) = \frac{-1}{s^2 + c/m \cdot s + k/m} = \frac{-1}{s^2 + 100/20 \cdot s + 4000/20}$$

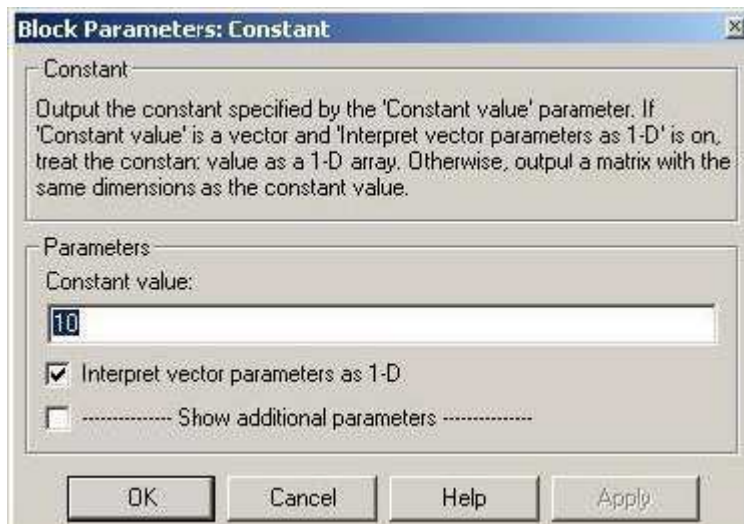
Bild 5.4a Parametrisierung des Blocks „Transfer Fcn“



$$u(t) = g + 1/m \cdot F(t).$$

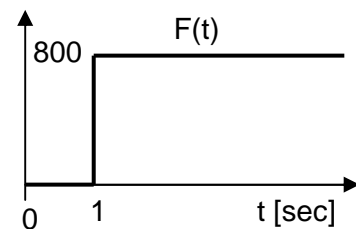
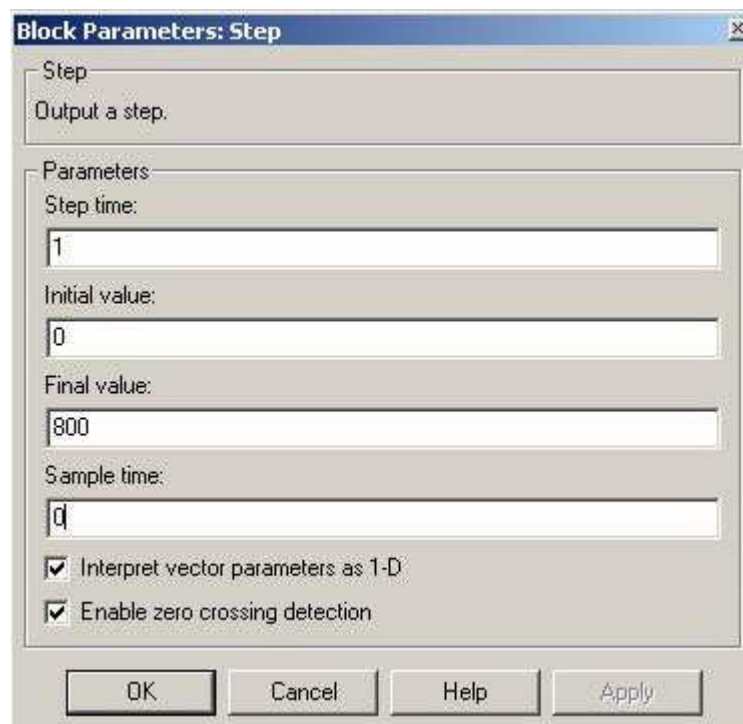
Verstärkung $1/m = 1/20$

Bild 5.4b Parametrisierung des Blocks „Gain“



$$g = 10$$

Bild 5.4c Parametrisierung des Blocks „Constant“



Kraftwirkung $F = 800\text{N}$

Bild 5.4d Parametrisierung des Blocks „Step“

Das SIMULINK Programm nach der Parametrisierung kann schon laufen und ist auf dem Bild 5.5 zu sehen. Um den Eingang graphisch darzustellen, wird ein zweiter Scope-Block aktiviert.

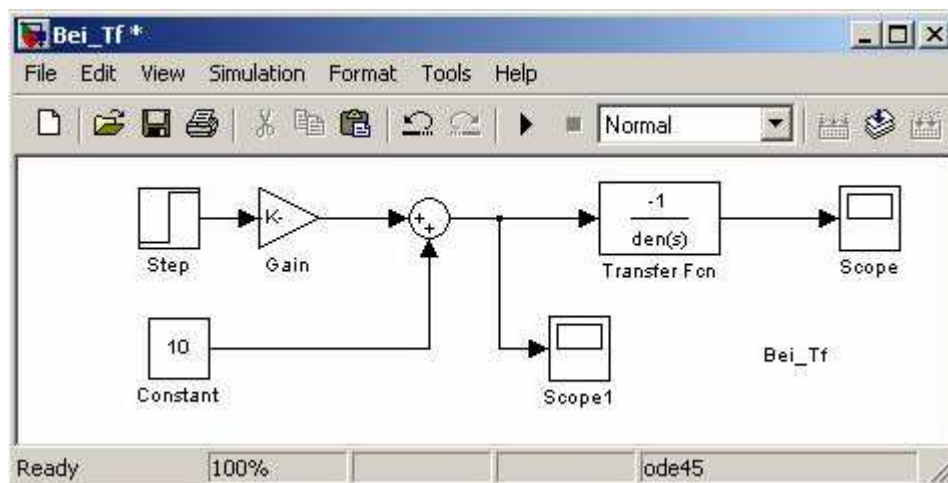
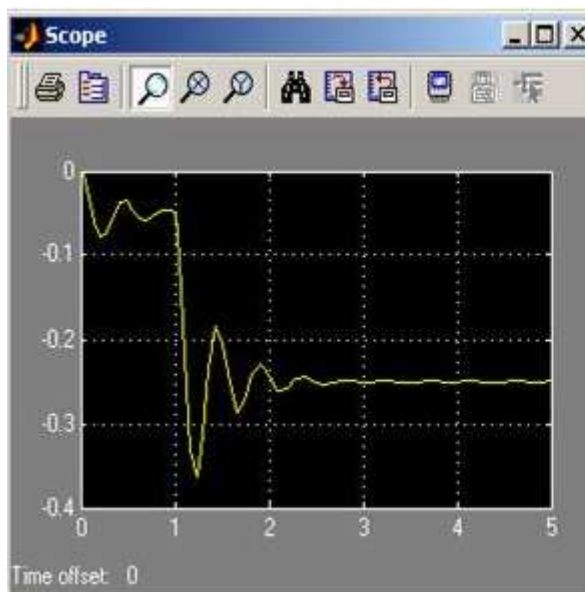
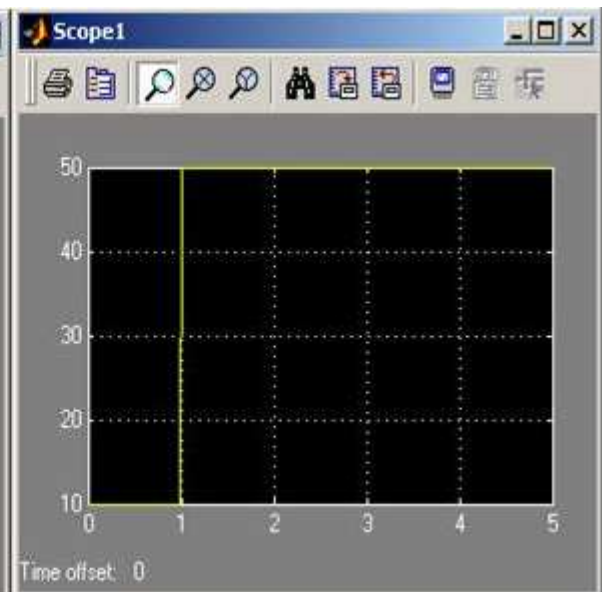


Bild 5.5 SIMULINK Programm nach der Parametrisierung

Die Simulationsergebnisse – der zeitliche Verlauf des Ausgangs $y(t)$ und des Eingangs $u(t)$ ist auf dem Bild 5.6,7 zu sehen.

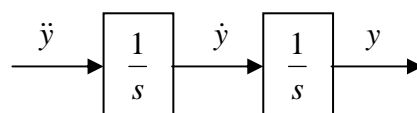
Bild 5.6 Zeitverlauf von $y(t)$ Bild 5.7 Zeitverlauf von $u(t)$

5.2.2. AUFSTELLUNG EINES SIMULINK PROGRAMMS, WENN DIE BESCHREIBUNG DURCH DIFFERENTIALGLEICHUNG GEGEBEN WIRD

Die Grundlage der Modellierung einer Differentialgleichung besteht an der Eigenschaften des Blocks „Integrator“. Es gilt, wenn ein Signal $x^n(t)$ durch den Block „Integrator“ geht, dann an dem Ausgang des Integrators hat das Signal die Ableitung um Eins kleiner, der Ausgang gleicht $x^{n-1}(t)$. Wir nehmen an, dass aus der Gleichung (5-1) die höchste Ableitung berechnet werden kann

$$\ddot{y} = -\frac{c}{m} \cdot \dot{y} - \frac{k}{m} \cdot y - g - \frac{F}{m} \quad (5-4)$$

Durch zwei Integratoren werden die Ableitung auf Null gesetzt (siehe Bild 5.8), was bedeutet, daß hinter dem zweiten Integrator das gesuchte Signal $y(t)$ ist.

Bild 5.8 Wirkung des Blocks „Integrator“ auf den Signal \ddot{y}

Die zweite Ableitung wird aus der Gleichung (5-4) durch Summation der rückgekoppelten Ableitungen \dot{y} , y über „Gain“ Blöcke c/m , k/m und der Eingangssignale g , $F(t)$ über „Gain“ Block $1/m$ gebildet.

Dies kann verallgemeinert werden auf eine lineare zeitinvariante Differentialgleichung n- Ordnung, wo n- Integratoren eingesetzt werden.

Ein SIMULINK Programm, das mit Hilfe dieses Verfahren aufgebaut ist, zeigt das Bild 5.9. Hier wurde gezeigt, wie die Blöcke „Multiplex, To File, To Workspace“ angewendet können.

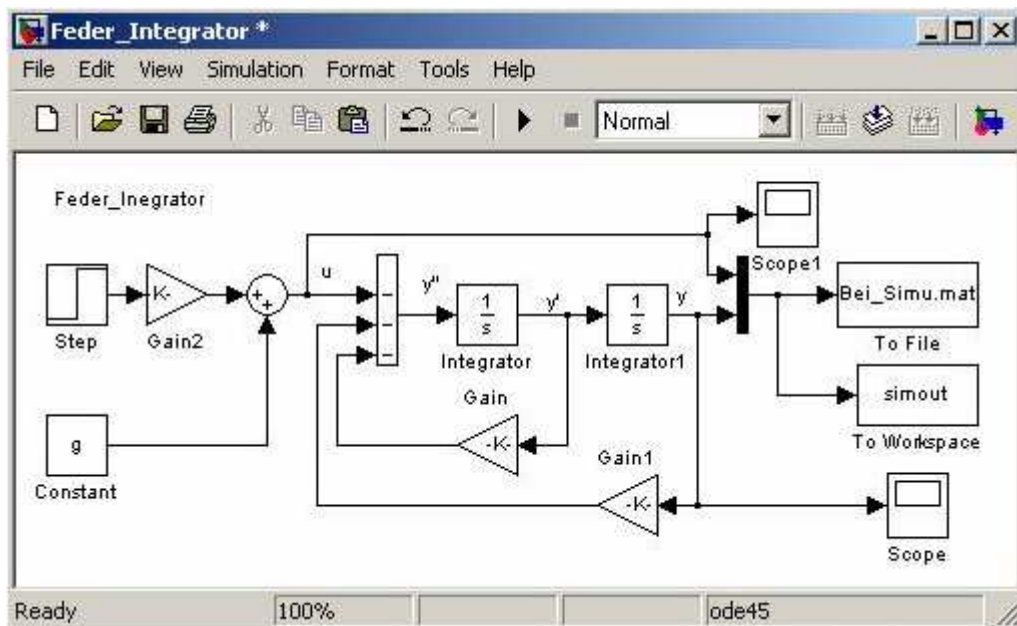


Bild 5.9 SIMULINK Programm des Federsitzmodells mit „Integrator, Multiplex, To File und To Workspace“

Bei Parametrisierung der Blöcke „Step, Constant, Gain, Gain1, Gain2“ können die Variablen in einem MATLAB Programm definiert werden, das auf dem Bild 5.10 dargestellt ist.

```

1 % Beispiel
2 display('Beispiel MATLAB Programm-Parametern Federsitz')
3 g=10
4 F=800
5 k=4000
6 c=100
7 m=20
  
```

Bild 5.10 Ein MATLAB Programm mit Parametern des Federsitzmodell

5.3. WIRKUNG DES FAHRERS

Die Gleichung (5-1) ist eine Beschreibung eines linearen zeitinvarianten Systems. Wir werden jetzt die Wirkung des Fahrers in Richtung des mathematischen Modells analysieren. Wenn sich der Fahrer, der die Masse $m_f = 80 \text{ kg}$ hat, auf den Federsitz hinsitzt, wird Folgendes aufgerufen:

- 1) Durch die Wirkung der Fallbeschleunigung g , wird eine Kraft von $F_f = g \cdot m_f = 800 \text{ N}$ auf den Sitz wirken
- 2) Die Masse des Fahrers addiert sich zu der Masse des Federsitzes und ist zeitlich abhängig von dem wann und wie lange der Fahrer auf dem Sitz ist. Für die Masse kann folgende Gleichung geschrieben werden.

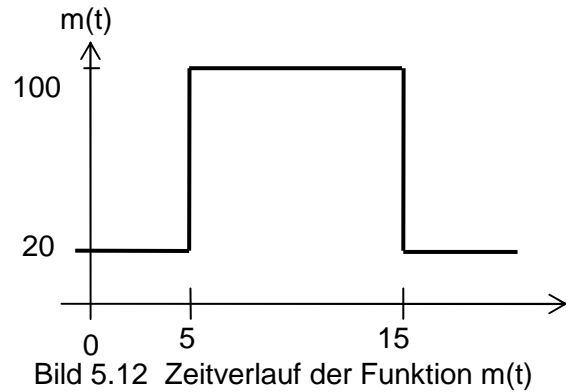


Bild 5.12 Zeitverlauf der Funktion $m(t)$

$$m(t) = m + m_f(t) \quad (5-5)$$

Der Zeitverlauf der Masse $m(t)$ ist auf dem Bild 5.8 skizziert. Weil die Masse zeitabhängig ist, muss diese in die Gleichung (5.1) eingeführt werden. Die Gleichung hat dann folgende Form

$$m(t) \cdot \ddot{y}(t) = -c \cdot \dot{y}(t) - k \cdot y(t) - m(t) \cdot g - F(t), \quad (5-6)$$

Wenn die Masse gleich $m(t) = m + m_f(t)$ ist, dann die Differentialgleichung (5.6) kann umgeschrieben werden

$$[m + m_f(t)] \cdot \ddot{y}(t) = -c \cdot \dot{y}(t) - k \cdot y(t) - [m + m_f(t)] \cdot g - F(t).$$

Die höchste Ableitung ist dann gleich

$$\ddot{y} = -\frac{c}{m(t)} \cdot \dot{y} - \frac{k}{m(t)} \cdot y - g - \frac{F}{m(t)} = -c \cdot \frac{\dot{y}(t)}{m(t)} - k \cdot \frac{y(t)}{m(t)} - g - \frac{F(t)}{m(t)}. \quad (5-7)$$

Aus dieser Gleichung ist es klar, daß die Signale $\dot{y}(t)$, $y(t)$, $F(t)$ durch die Koeffizienten multipliziert und durch das Signal $m(t)$ dividiert werden. Hier wird der Block „Product“, der eine Multiplikation oder Dividierung von Signalen ermöglicht, benutzt. Weiter werden die Signale $m(t)$ und $F(t)$ als Signale im SIMULINK Programm modelliert werden.

Parametrisierung des Blockes „Product“

In der „Simulink Library Bibliothek“ wird durch Doppelklick die Bibliothek „Math Operations“ geöffnet. Das Symbol des „Product“ Blockes wird in das eigenes Fenster des SIMULINK Programms gezogen (siehe Bild 5.13a). Durch anklicken des Symbols öffnet sich Parametrierungsfenster „Block Parameters:Product“, siehe Bild 5.13b. In dem Feld „Number of inputs“ kann man entweder eine Zahl der Eingänge eingeben, oder durch Symbole „*“, „/“ festlegen, welcher Eingang multipliziert und welcher dividiert wird. Wir haben festgelegt, erster wird multipliziert, der zweite wird dividiert.

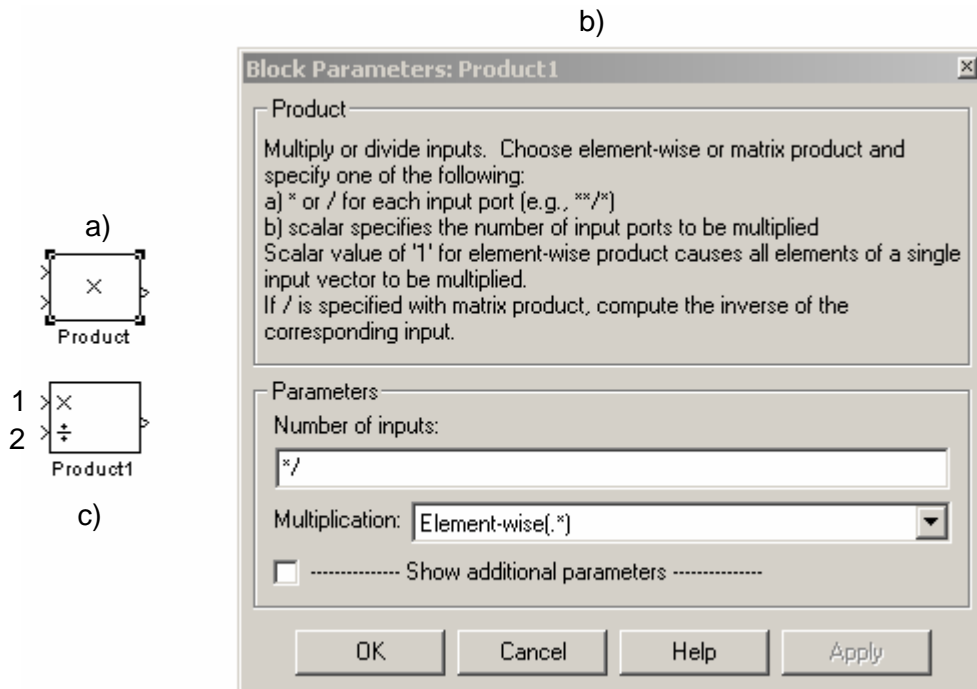


Bild 5.13 Parametrisierung des Blocks „Product“

Im Bild 5.14 ist das Programm dargestellt. Die Simulationszeit wird auf 20sec eingestellt.

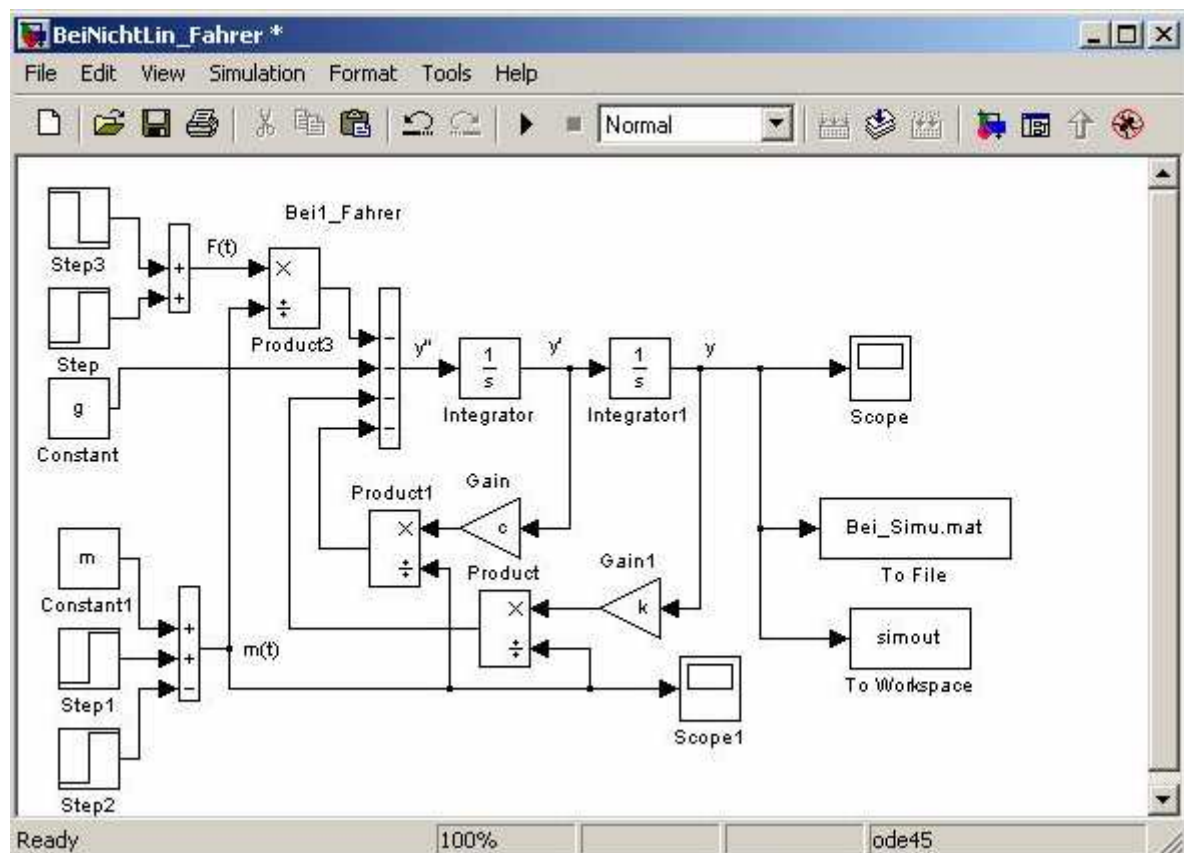
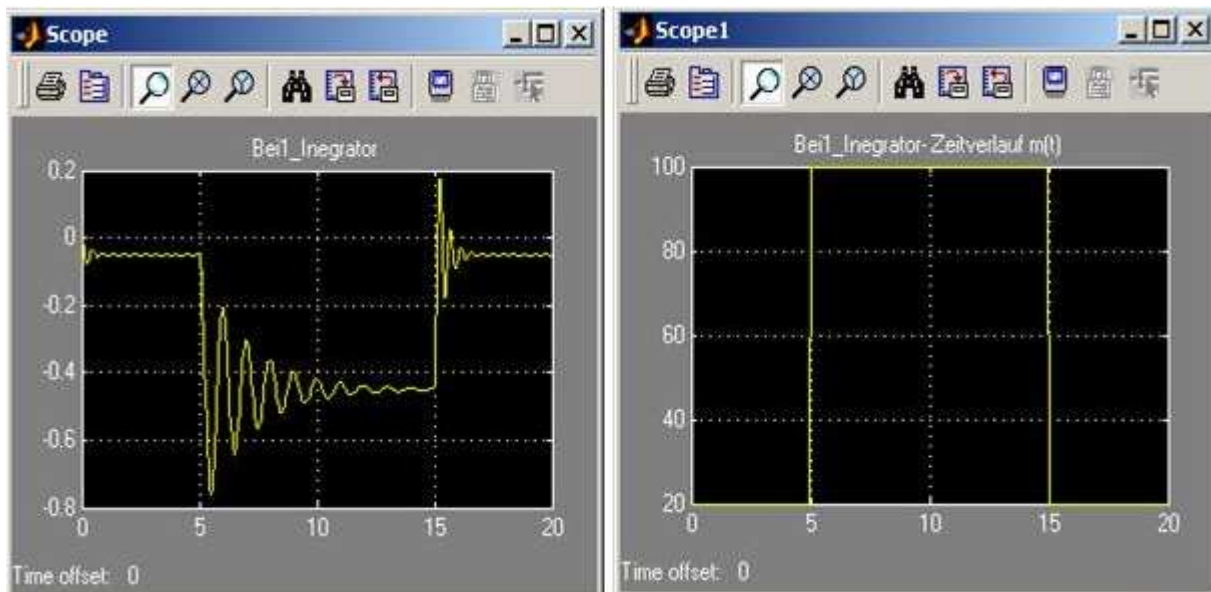


Bild 5.14 SIMULINK Programm – simuliert Wirkung eines Fahrers auf die Bewegung des Federsitzes

Bild 5.15 a) Zeitverlauf von $y(t)$ b) Zeitverlauf von $m(t)$

Aus dem Bild 5.15a folgt, dass die Wirkung des Massenfahrs in die Frequenz der Schwingung hat sich ausgewirkt. Die Frequenz des Federsitzes ist kleiner, wenn der Fahrer auf dem Sitz ist, als wenn der Fahrer den Sitz verlässt.

5.4. WIRKUNG UND MODELLIERUNG DER KOEFFIZIENT $c = c(\dot{y})$

Die Anwendung der „Fcn“ Blöcke wird am meisten bei Lösungen einer nichtlinearen Differentialgleichung angewendet. Wir möchten die Anwendung an folgendem Beispiel demonstrieren. Wir nehmen an, dass die Differentialgleichung (5-4) des Federsitzes hat die Form,

$$\ddot{y} = -\frac{c(\dot{y})}{m} \cdot \dot{y} - \frac{k}{m} \cdot y - g - \frac{F}{m}, \quad (5-8)$$

wo der Koeffizient der Dämpfung eines neuen Dämpfers folgend definiert ist:

$$c(\dot{y}) / m = 5 - 50 * \text{abs}(\dot{y}) + 20 * (\dot{y})^2. \quad (5-9)$$

Bei dem Programmaufbau gehen wir von dem SIMULINK Programm im Bild 5.9 aus. Der Koeffizient c wird für diesen Fall keine Konstant sein, aber es wird für die Berechnung der Funktion (5-9) ein „Fcn“ Block benutzt werden. Das Programm ist auf dem Bild 5.16 dargestellt. Die Funktion (5-9) wird direkt bei der Parametrisierung des Blocks „Fcn“ geschrieben. Der Eingang des Blocks ist die Ableitung \dot{y} , der Ausgang ist der berechneten Wert von der Funktion $c(\dot{y})$. Zeitverläufe von u , y sind im Bild 5.17 zu sehen.

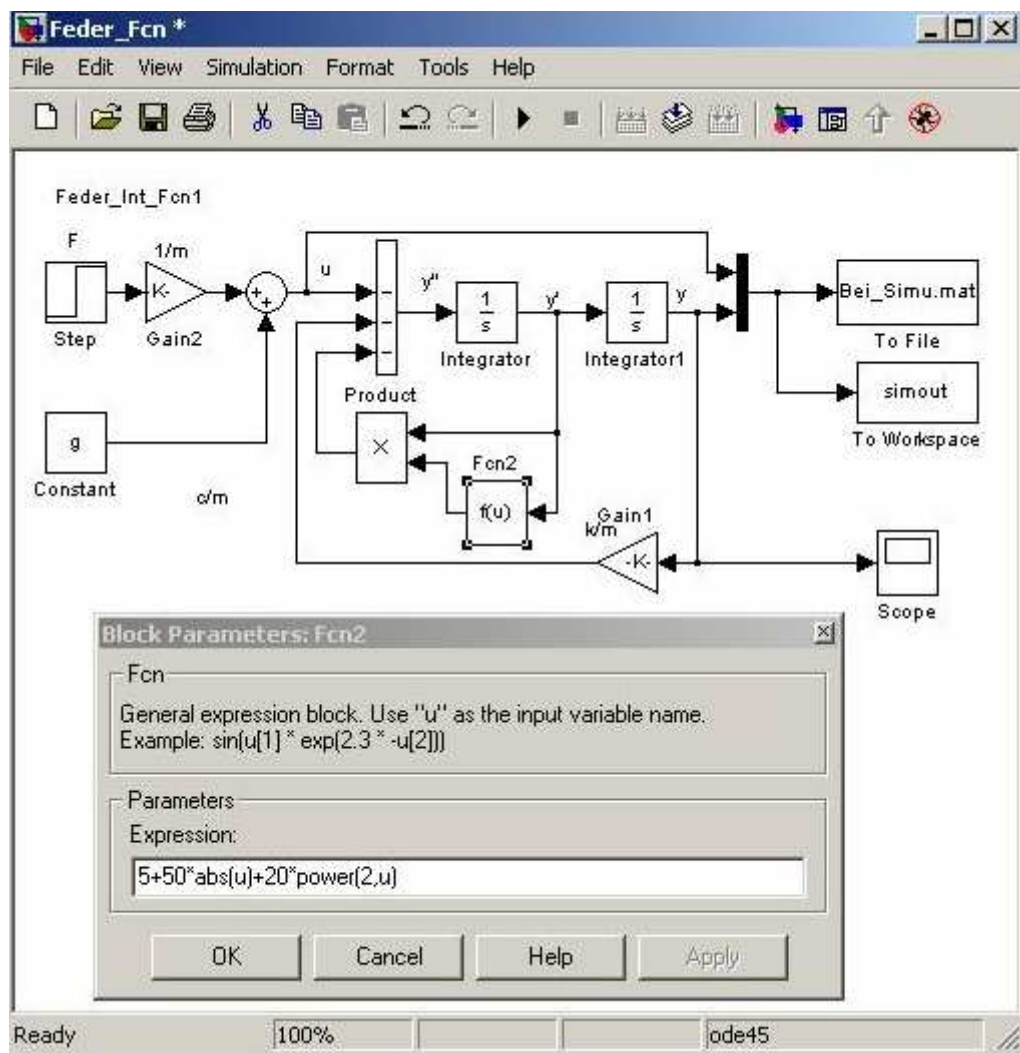
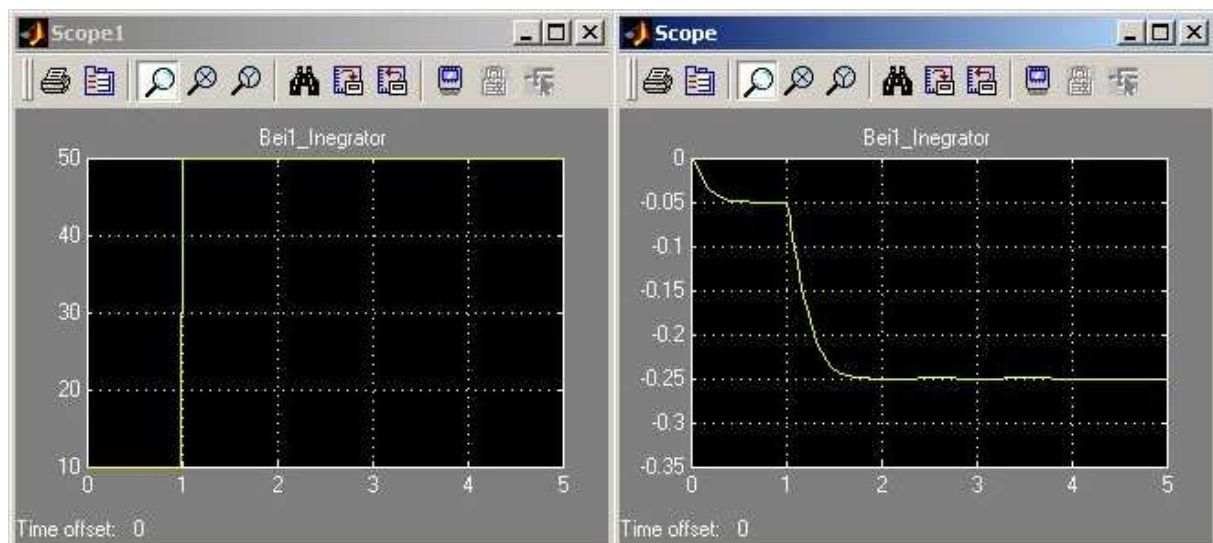


Bild 5.16 Anwendung des „Fcn“ Blocks.

Bild 5.17 Zeitverlaufs de Eingangs und Ausgangs des Federsites mit $c = c(y')$

6. BLOCKPARAMETRISIERUNG

6.1.PARAMETRIERUNG DER SIMULINK- SIMULATION

Nach Parametrierung der Blöcke des Modells kann die Parametrierung der gesamten Simulationszeit durchgeführt werden. In der Schaltfläche des SIMULINK Modells klickt man auf Simulation und es öffnet sich ein Rollmenü, siehe Bild 6a. Durch Klicken auf die Option „Parameters“ öffnet sich das Dialogfenster „Simulation Parameters: untitled“ (siehe Bild 6b).

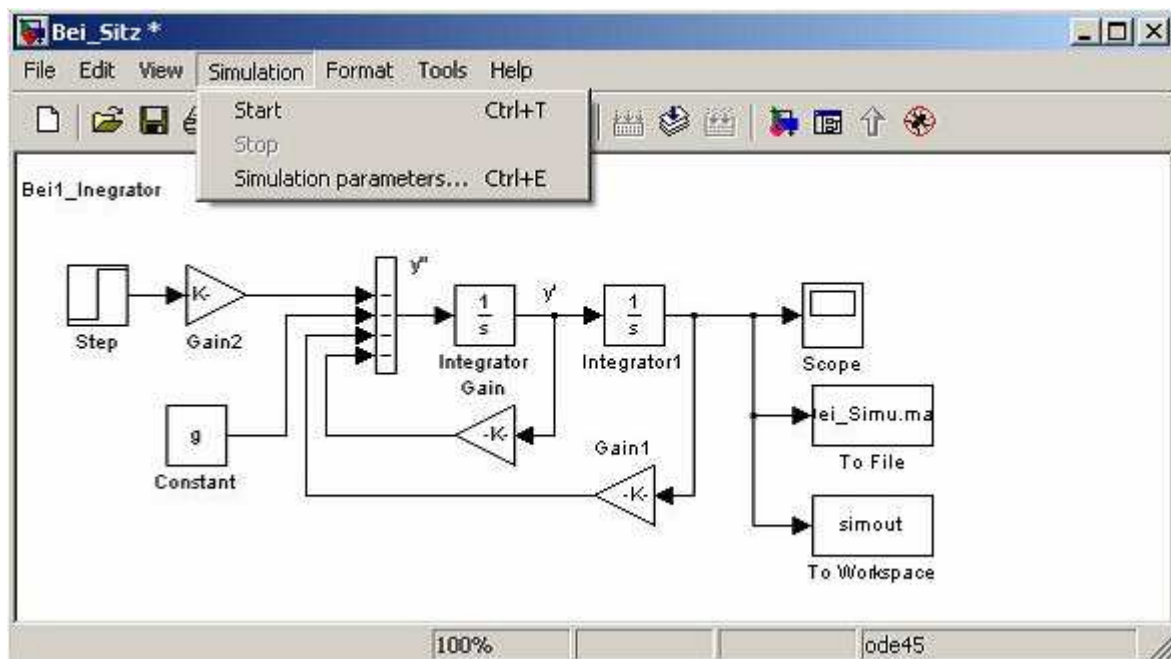


Bild 6.1a Das Rollmenü der Simulation

Dialog Fenster „Simulation Parameters“ enthält:

- | | |
|------------------------|-----------|
| 1) Solver- Fenster | Bild 6.1b |
| 2) Workspace- Fenster | Bild 6.2 |
| 3) Diagnostik- Fenster | Bild 6.3 |
| 4) Advanced- Fenster | Bild 6.4 |

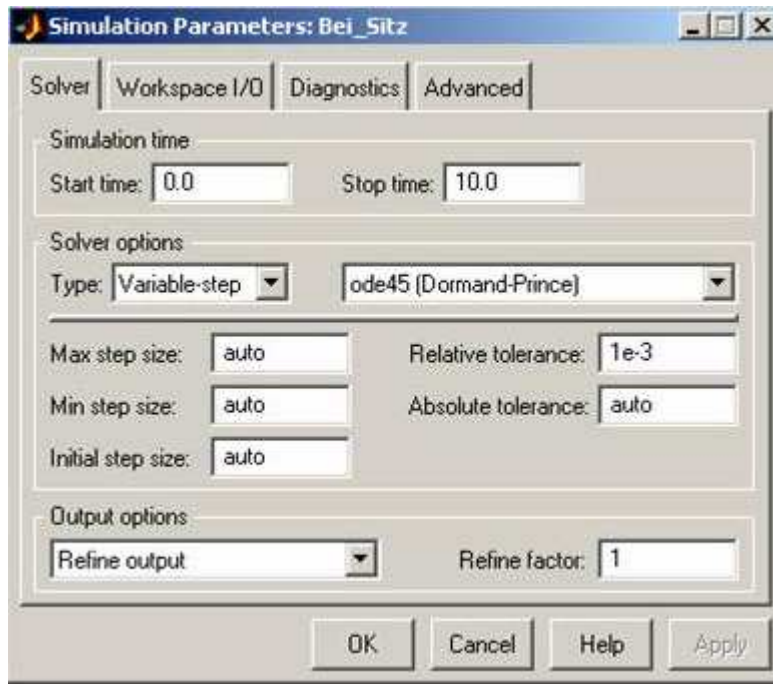


Bild 6.1b Das Dialogfenster „Simulation Parameters: Bei_Sitz

1) Im Solver- Fenster wird eingestellt

- die Start und -Stoppzeit,
- Type des Integrationsschritts **Variable-step** oder **Fix-step**
- Integrationsverfahren (**ode45**, **Runge-Kutta 5.Ordnung**)
- Die Max Step size und Initial size sollten auf **auto** gelassen
- Output option soll als „**Refine output**“ gewählt
- Refine factor soll auf 1 eingestellt werden

2) Workspace- Fenster

Hier werden Input und Output Variabel dargestellt (Bild 6.2.)

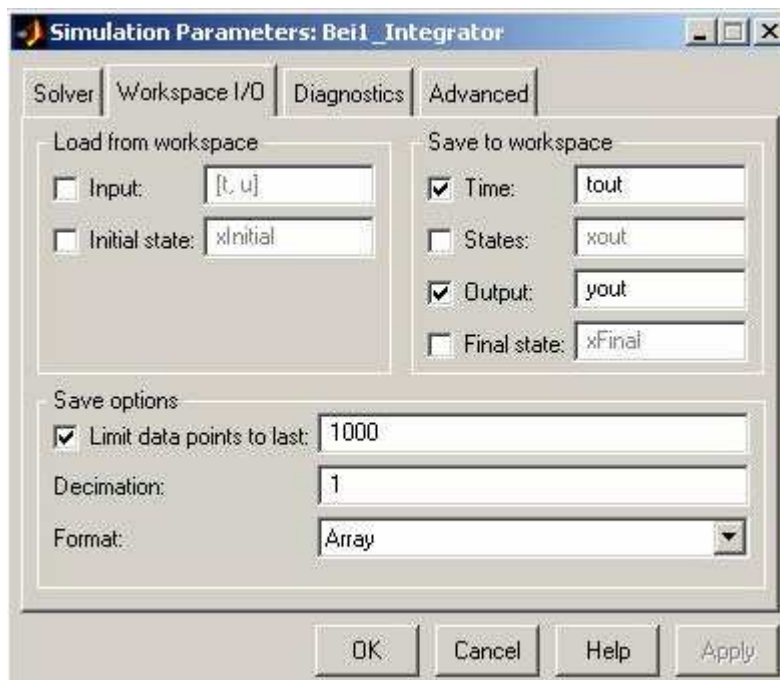


Bild 6.2 Workspace-Fenster

3) Diagnostic- Fenster

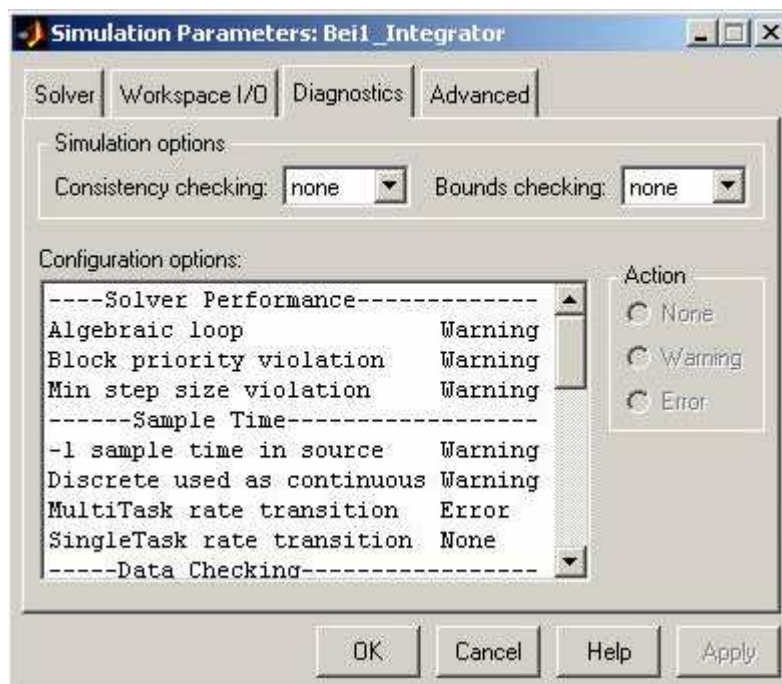


Bild 6.3 Diagnostic-Fenster

4) Advanced- Fenster

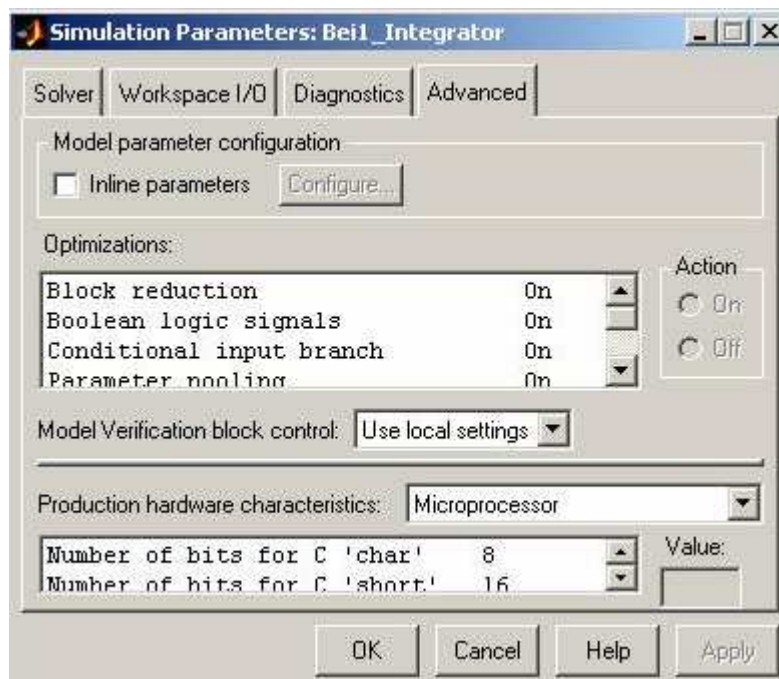


Bild 6.4 Advanced-Fenster

6.2. PARAMETRIERUNG DES BLOCKS „CONSTANT“

Durch Doppelklick wird der Block „Constant“ geöffnet und die Parametrierung erfolgt durch Eingeben des Wertes (siehe Bild 6.5).

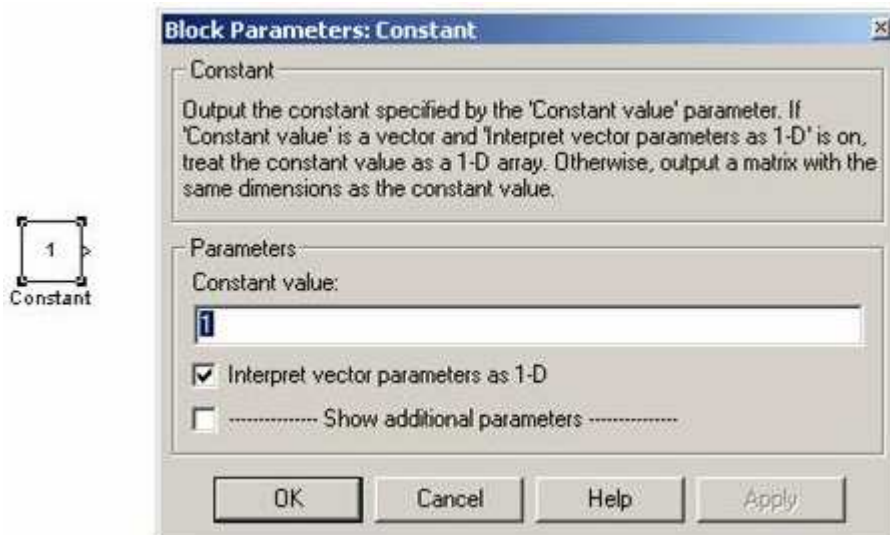


Bild 6.5 Dialogfenster zur Parametrierung des Blocks „Constant“

6.3. PARAMETRIERUNG DES BLOCKS „DISPLAY“

Durch Doppelklick wird der Block „Display“ geöffnet und die Parametrierung erfolgt durch Eingeben des Wertes (siehe Bild 6.6).

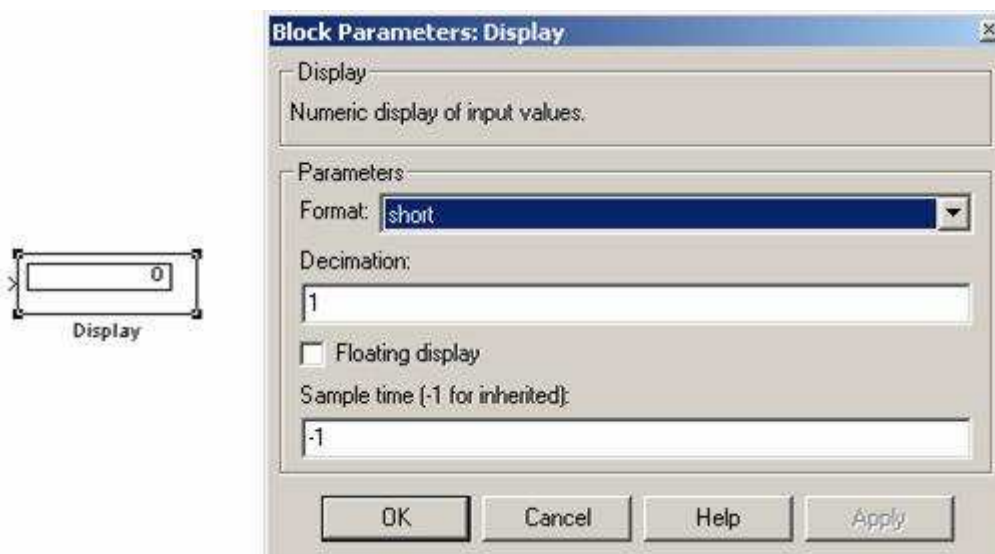


Bild 6.6 Dialogfenster zur Parametrierung des Blocks „Display“

6.4. PARAMETRIERUNG DES BLOCKS SLIDERGAINS

Durch Doppelklick auf den Block „Slider Gain“ wird das Fenster „Slider Gain“ geöffnet (siehe Bild 6.7). Das Verändern der Ausgangsgröße des „Slider Gain“ erfolgt durch Anklicken der Flächen links/rechts zwischen dem Schieber und den Endmarken des Schiebereiches.



Bild 6.7 Das Dialogfenster zur Einstellung der Ausgangsgröße des Blocks „Slider Gain“

6.5. PARAMETRIERUNG DES BLOCKS STEP

Durch Doppelklick auf den Block „Step“ wird das Fenster Bild 6.8 geöffnet. In dem Felder „Step Time:“ wird die Zeit des Stepeingangs, im „Initial value“ der Anfangswert, im „Final value:“ der Endwert und im „Sample time“ die Abtastperiode wird eingegeben.

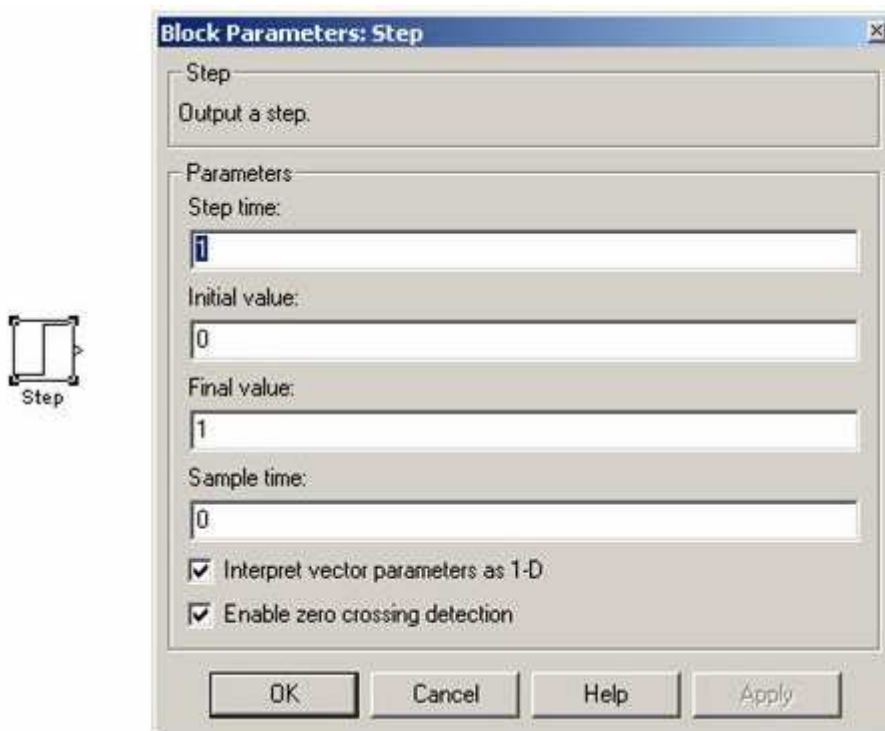


Bild 6.8 Parametrierung Fenster des Blocks „Step“

6.6. PARAMETRIERUNG DES BLOCKS „SCOPE“

Durch Doppelklick auf den Block „Scope“ wird das „**Scope**“-Fenster geöffnet. Wenn das Fenster offen ist, bekommt man den Verlauf der Größen, die in dem Block den Eingang bilden (Bild 6.9a), dargestellt.

Hinweis: Eine Darstellung findet nur in einem vorher in „**Display**“ und „**Scope**“ eingestellten Zeitbereich statt. Durch das Schließen von Fenstern und Bildern werden alle Messdaten gelöscht.

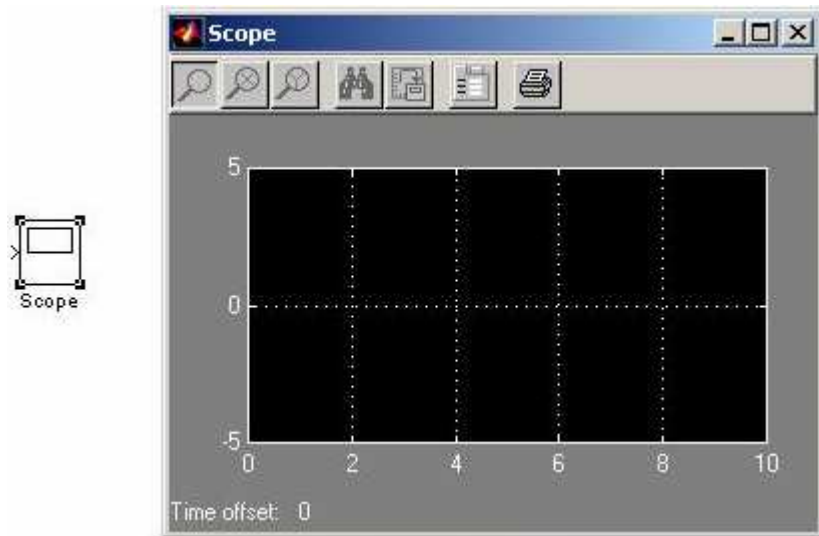


Bild 6.9a Scope – Fenster



Bild 6.9b Scope properties Fenster

Mit der Schaltfläche „**Fernglas**“ kann eine automatische Skalierung durchgeführt werden. Es sind 3 Zoom-Funktionen vorhanden. Durch einen Klick mit der rechten Maustaste auf die Bildfläche wird das Menü Bild 6.9b geöffnet, von dem die „**Axis properties**“ geöffnet werden kann. Eine Beschriftung kann in die Zeile „**Title**“ eingegeben werden.

6.7. PARAMETRIERUNG DES BLOCKS „TO FILE“

Durch Doppelklick auf den Block „To File“ wird das Fenster „**Block Parameters: To File**“ geöffnet, siehe Bild 6.10. In dem Feld „**Filename**“ wird der Dateiname, im Feld „**Variable Name**“ die Variable, im Feld „**Decimation**“ der Decimation-Faktor (Voreingestellt auf 1) und im Feld „**Sample Time**“ die Abtastperiode eingegeben.

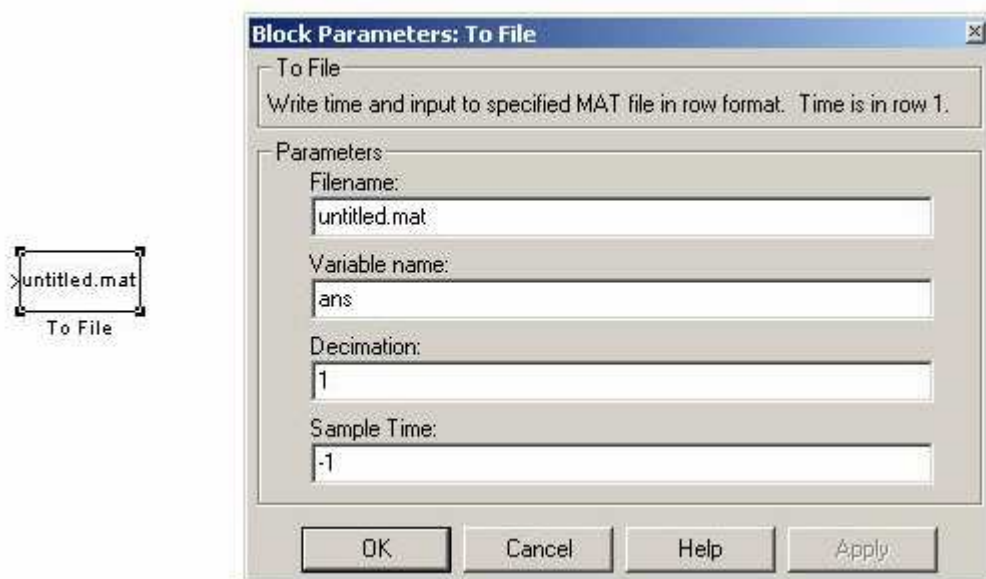


Bild 6.10 Das Fenster für Parametrierung des Blocks „To File“

6.8. PARAMETRIERUNG DES BLOCKS „PID CONTROLLER“

Durch Doppelklick auf den Block „PID Controller“ wird das Fenster „**Block Parameters : PID Controller**“ geöffnet, siehe Bild 6.11. Die Übertragung des Reglers ist: $R(s) = P + I/s + Ds$.

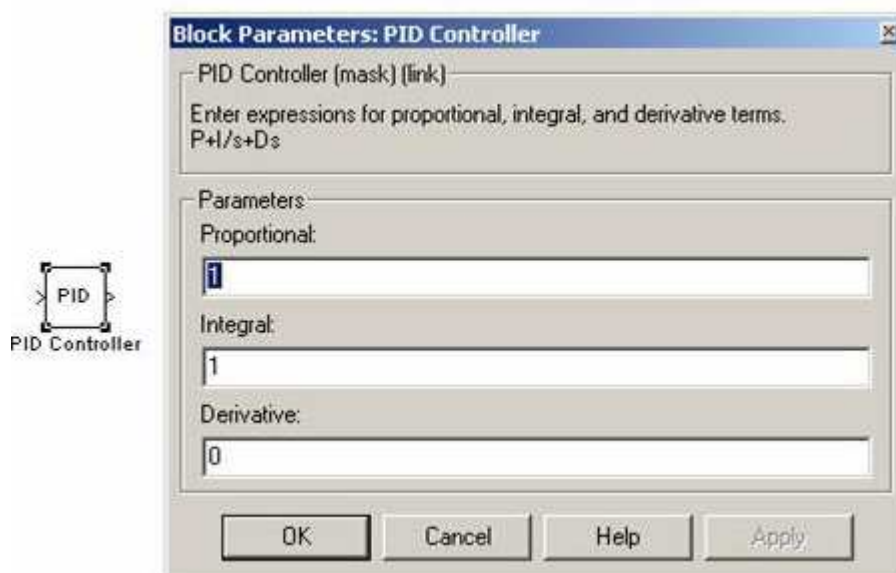


Bild 6.11 Fenster für Parametrierung des Blocks „PID Controller“

6.9. PARAMETRIERUNG DES BLOCKS „PID CONTROLLER WITH APPROXIMATE DERIVATION“

Durch Doppelklick auf den Block „**PID Controller (with Approximate Derivative)**“ wird das Fenster „Block Parameters: PID Controller (with Approximate Derivative)“ geöffnet, siehe Bild 6.12. Die Übertragung des Reglers ist: $R(s) = P + I / s + Ds / (Ns + 1)$

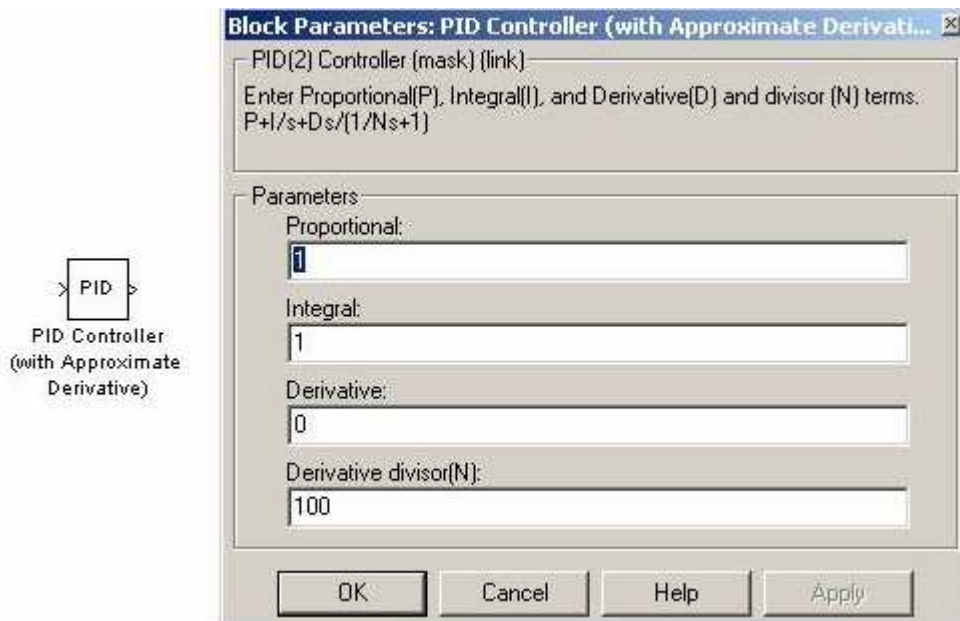


Bild 6.12 Fenster für Parametrierung des Blocks „PID Controller (with Approximate Derivative)“

6.10. PARAMETRIERUNG DES BLOCKS „TRANSFER FCN“

Durch Doppelklick auf den Block „**Transfer Fcn**“ wird das Fenster „**Block Parameters : Transfer Fcn**“ geöffnet (Bild 6.13). In das Feld Numerator/Denominator können direkt Parameter oder Namen der Polynome im Zähler und Nenner eingegeben werden.

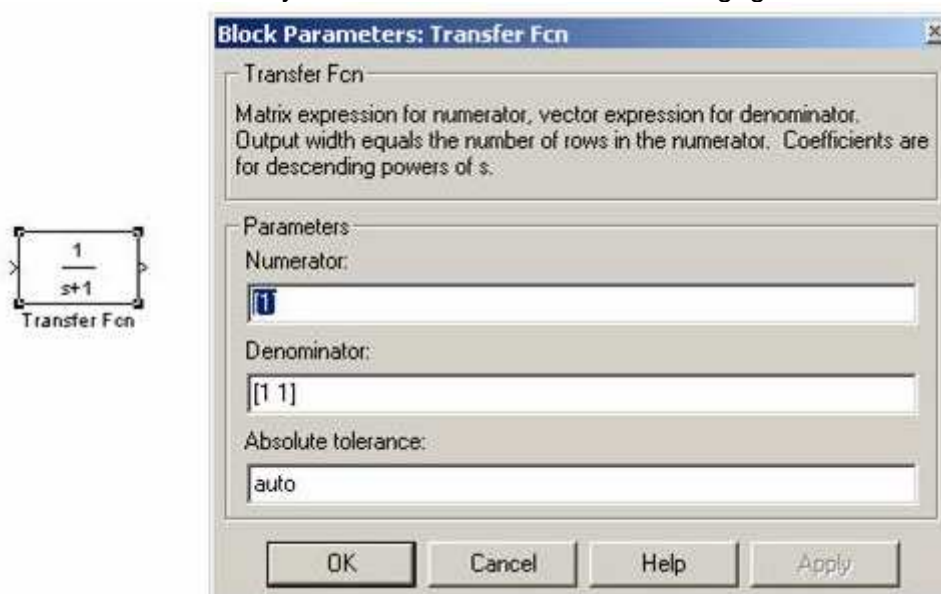


Bild 6.13 Parametrierungsfenster des Blocks „Transfer Fcn“

6.11. PARAMETRIERUNG DES BLOCKS „REPEATING SEQUENCE“

Durch Doppelklick auf den Block „Repeating Sequence“ wird das Fenster „**Block Parameters: Repeating Sequence**“ geöffnet, siehe Bild 6.14.

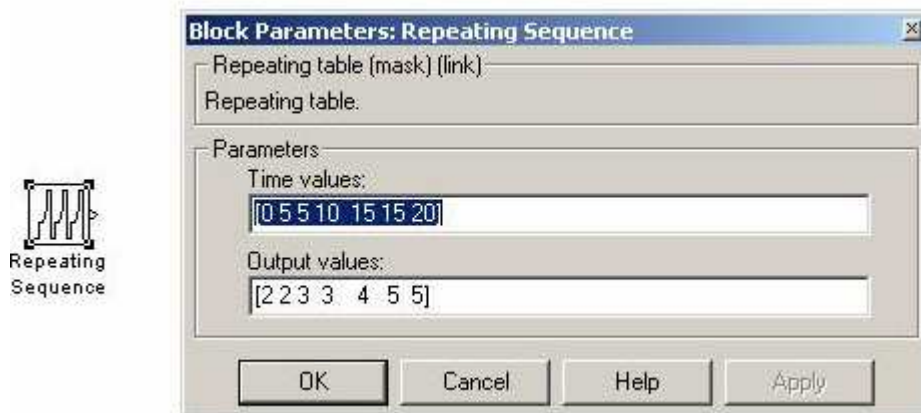


Bild 6.14 Fenster für Parametrierung des Blocks „Repeating Sequence“

Die eingestellten Parameter sichern den Ausgang nach dem Verlauf auf dem Bild 6.15.

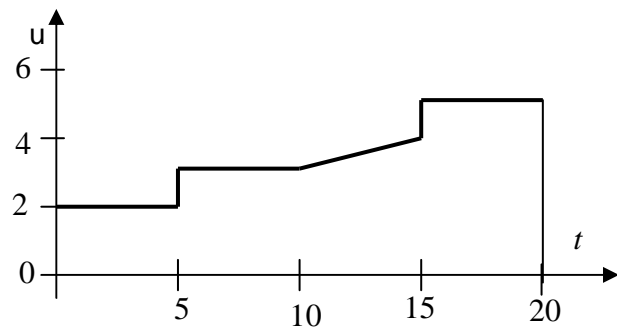


Bild 6.15 Ausgang

6.12. PARAMETRIERUNG DES BLOCKS „PRODUCT“

Durch Doppelklick auf den Block „Product“ wird das Fenster „**Block Parameters: Products**“ geöffnet, siehe Bild 6.16. In dem Feld „**Number of inputs**“ kann man entweder eine Zahl der Eingänge eingeben, oder durch Symbole „*“, „/“ festlegen, welcher Eingang wird multipliziert und welcher wird dividiert.

Durch Ankreuzen von „**Show additional parameters**“ ist es möglich noch weitere Parameter einzustellen.

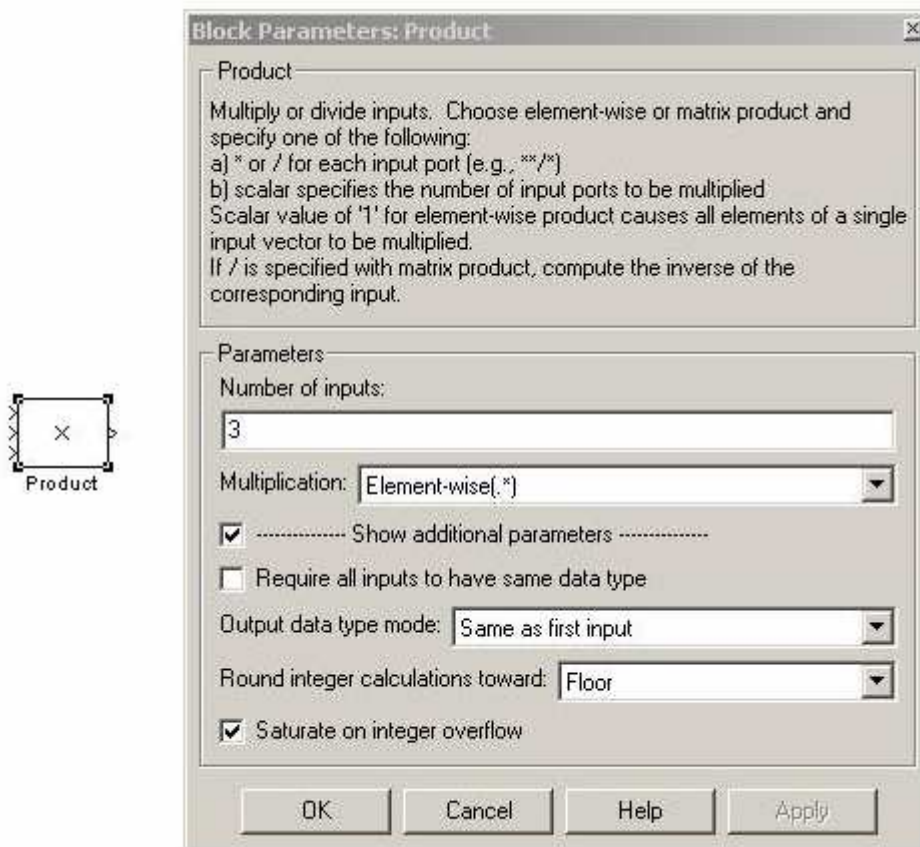


Bild 6.16 Parametrisierung des Blocks „Product“

6.13. PARAMETRIERUNG DES BLOCKS „Fcn“

Dieser Block berechnet durch die vorgeschriebene Funktion in dem „**Expression:**“-Fenster den Wert der Funktion für die Eingangsparameter $u(1)$, $u(2)$,.... Diese Funktion muss in der Struktur der C - Sprache geschrieben sein. Der Ausgang aus diesem Block ist immer nur ein Skalarprodukt. Die Funktion wird bei der Parametrisierung in das Feld „Expression:“ eingegeben!

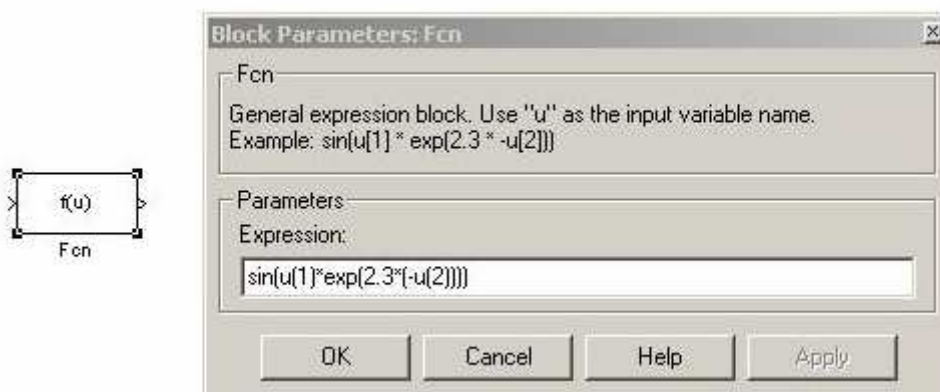


Bild 6.17 Parametrisierung des Blocks „Fcn“

6.14. PARAMETRIERUNG DES BLOCKS „MATLAB FCN“

Dieser Block ruft die definierte MATLAB Funktion aus dem Arbeitsspeicher auf und berechnet mit den Eingangsparametern die MATLAB Funktion. Der Ausgang oder Ausgänge aus diesem Block sind dann die berechneten Werte. Die MATLAB Funktion muss als eine MATLAB File –(Datei.m) geschrieben werden und in den Arbeitsspeicher zur Verfügung stehen!

Durch Doppelklick auf den Block öffnet sich das Fenster „**MATLAB Fcn**“, wo der Name der MATLAB Funktion mit Eingangsparameter „*sin(u)*, *foo(u(1),u(2))*“ oder ohne Parameter „*sin*“ eingegeben wird. Wenn die Funktion mehr Ausgänge hat, wird die Zahl der Ausgänge in das Fenster „**Output dimension**“ eingegeben.

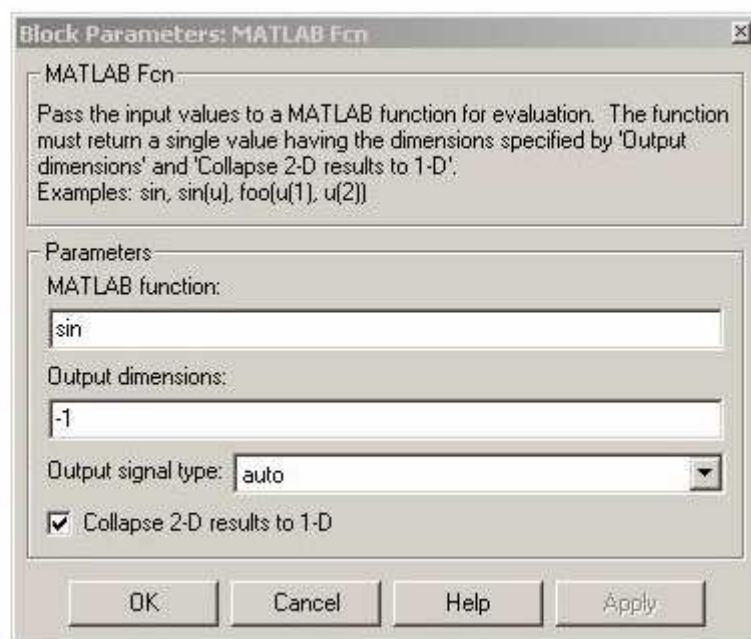
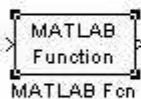


Bild 6.18 Parametrierung des Blocks „MATLAB Fcn“

Bemerkung:

Der Block „MATLAB Fcn“ ist wesentlich langsamer als der Block „Fcn“, weil bei jedem Integrationsschritt diese Funktion aufgerufen wird.

7. UNTERSYSTEME

Wenn das SIMULINK Programm durch die Anzahl der Blöcke zu groß und unübersichtlich ist, es ist Möglich durch Untersysteme eine Vereinfachung und Übersicht wieder zu erzielen. Wir zeigen das an unserem Modell des Federsitzes. Den Teil des Programms, den man als Untersystem einbinden will, wird mit der Maus in einem Viereck umgefahren und dadurch markiert (siehe Bild 7.1). Über das Menü „Edit“ wird durch das Anklicken der Funktion „Create subsystem“ gebildet (siehe Bild 7.2). Das Programm mit dem Untersystem ist im

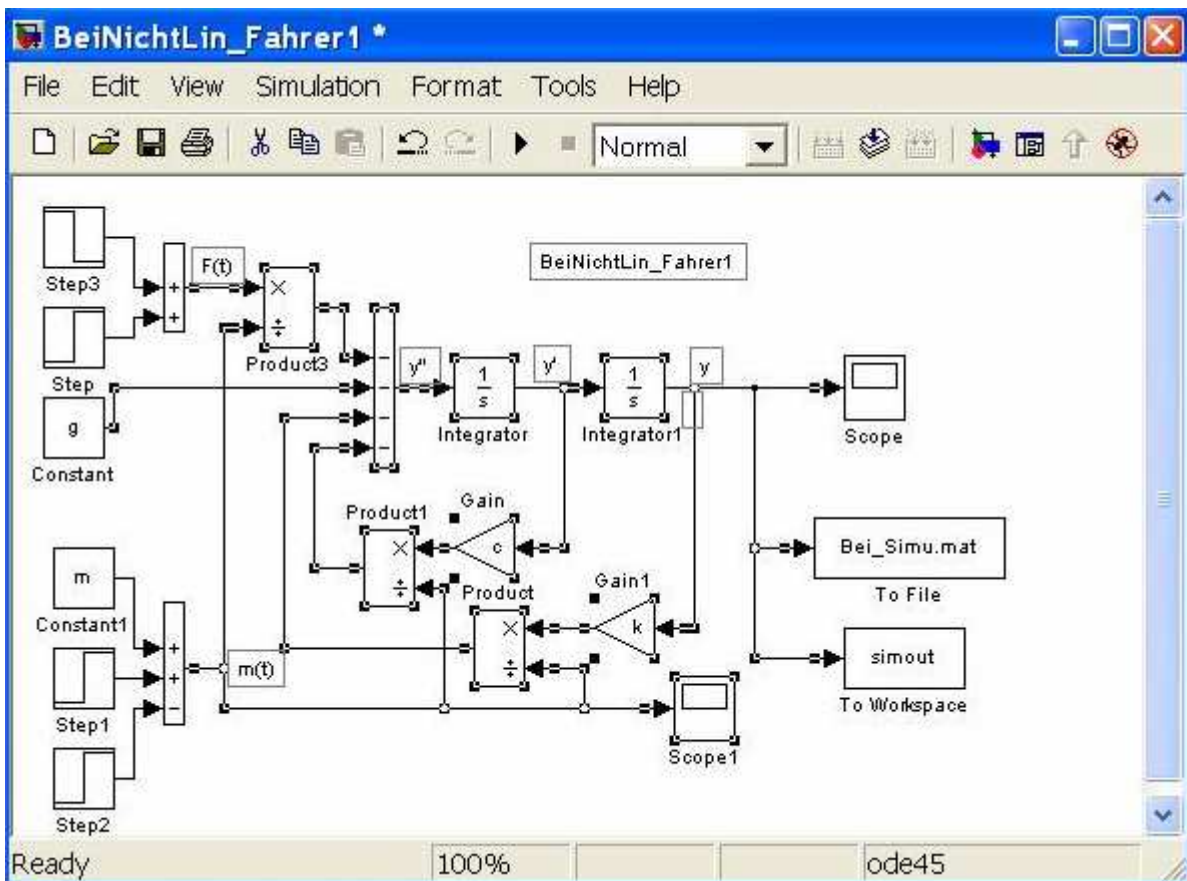


Bild 7.1 Teileinbindungen des Programms, das als Untersystem gebildet werden soll

Bild 7.2 demonstriert.

Durch ein Doppelclick an das Subsystem Symbol, öffnet sich ein separates Fenster, wo das Model des Subsystems dargestellt ist (siehe Bild 7.3). Aus dem Bild 7.2 sieht man, das das Subsystem 3 Eingänge und 1 Ausgang hat. Diesen Ein- und Ausgängen entsprechen im Bild 7.3 die 3 **Inports** aus und ein **Output** (Grenzen). Die Parameter des Untersystems können in dem Fenster .../Subsystem eingestellt werden.

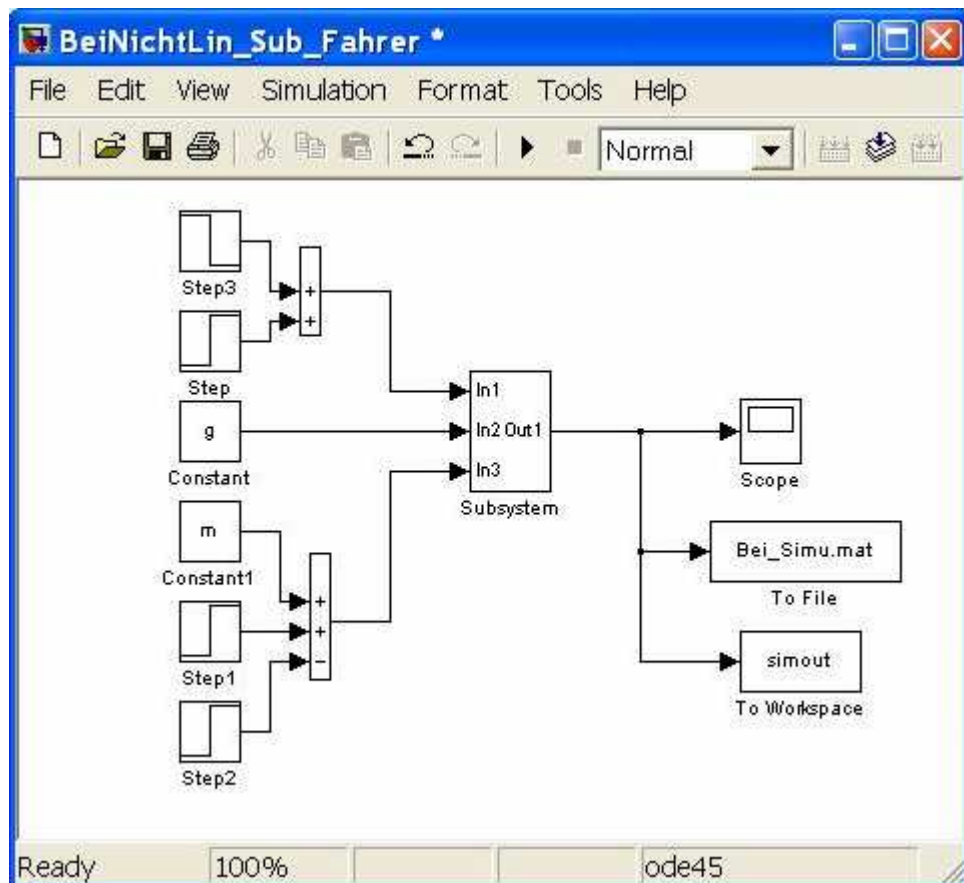


Bild 7.2 SIMULINK Modell mit dem Subsystem

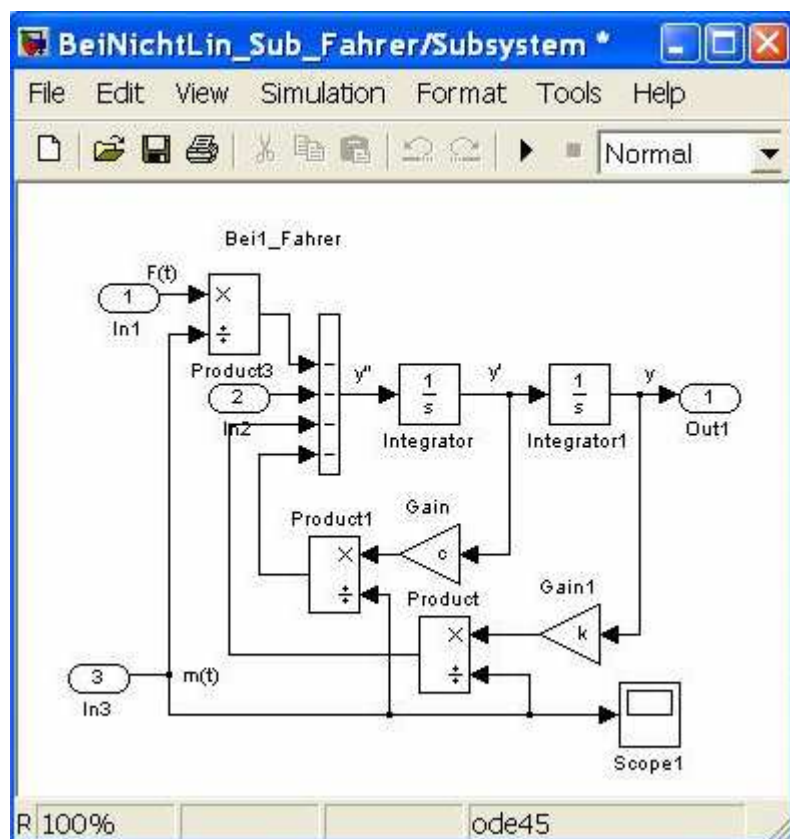


Bild 7.3 SIMULINK Modell von dem Subsystem

8. UNTERSUCHUNG DER REALEN OBJEKTE

Ein reales Objekt oder Anlage wird mittels „**real time toolbox**“ mit dem PC verbunden. Der Block „Adapter“ dient zur Konfiguration der Messkarte I/O. Dieser Block ist enthalten in Rollmenü „Real-Time Toolbox“, siehe Bild 8.1).

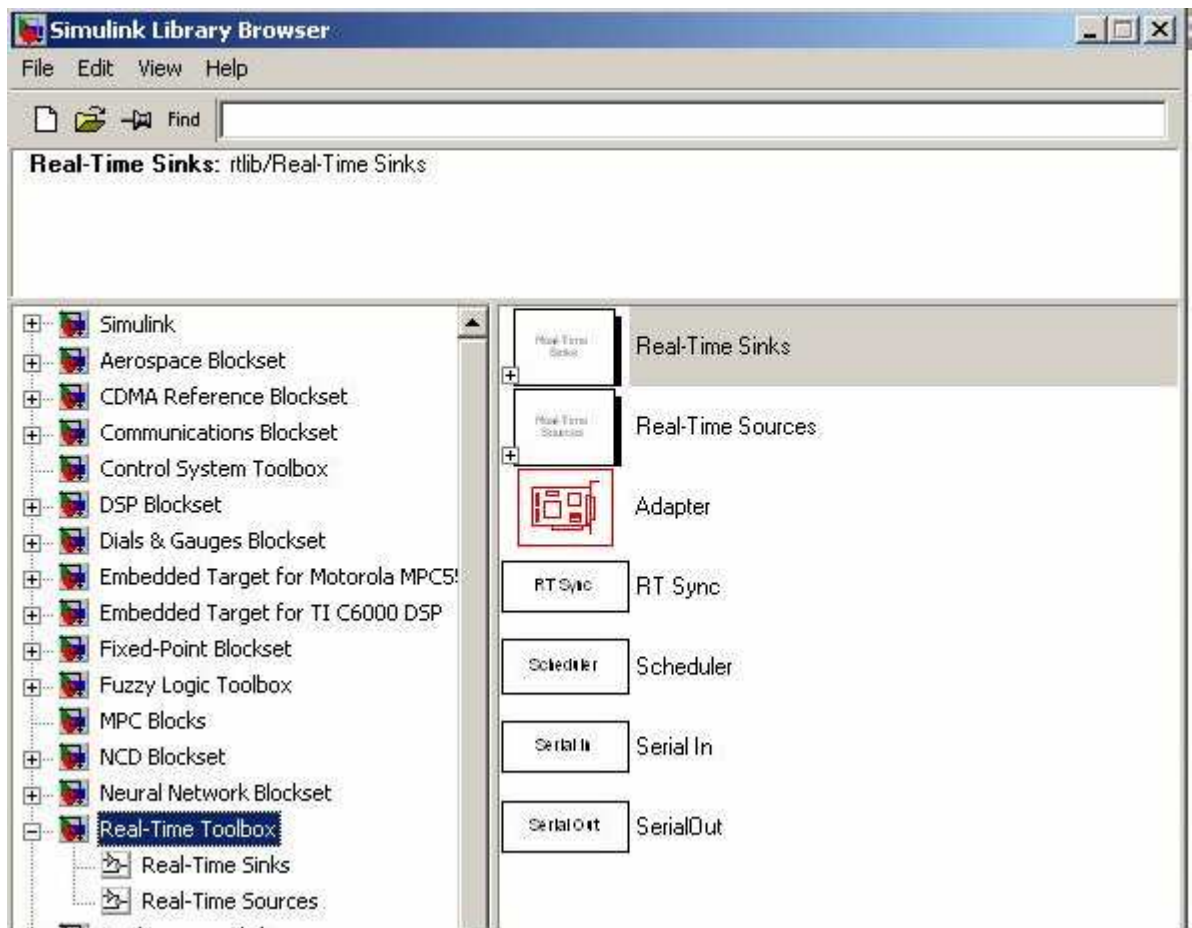


Bild 8.1: Bibliothek Real Time Toolbox

Wenn der Block nicht konfigurierbar ist, wählen Sie die Hardware aus der Liste. Weiter konfigurieren Sie die Hardware.

er noch
fenster „Se-
er I/O Karte
nach der

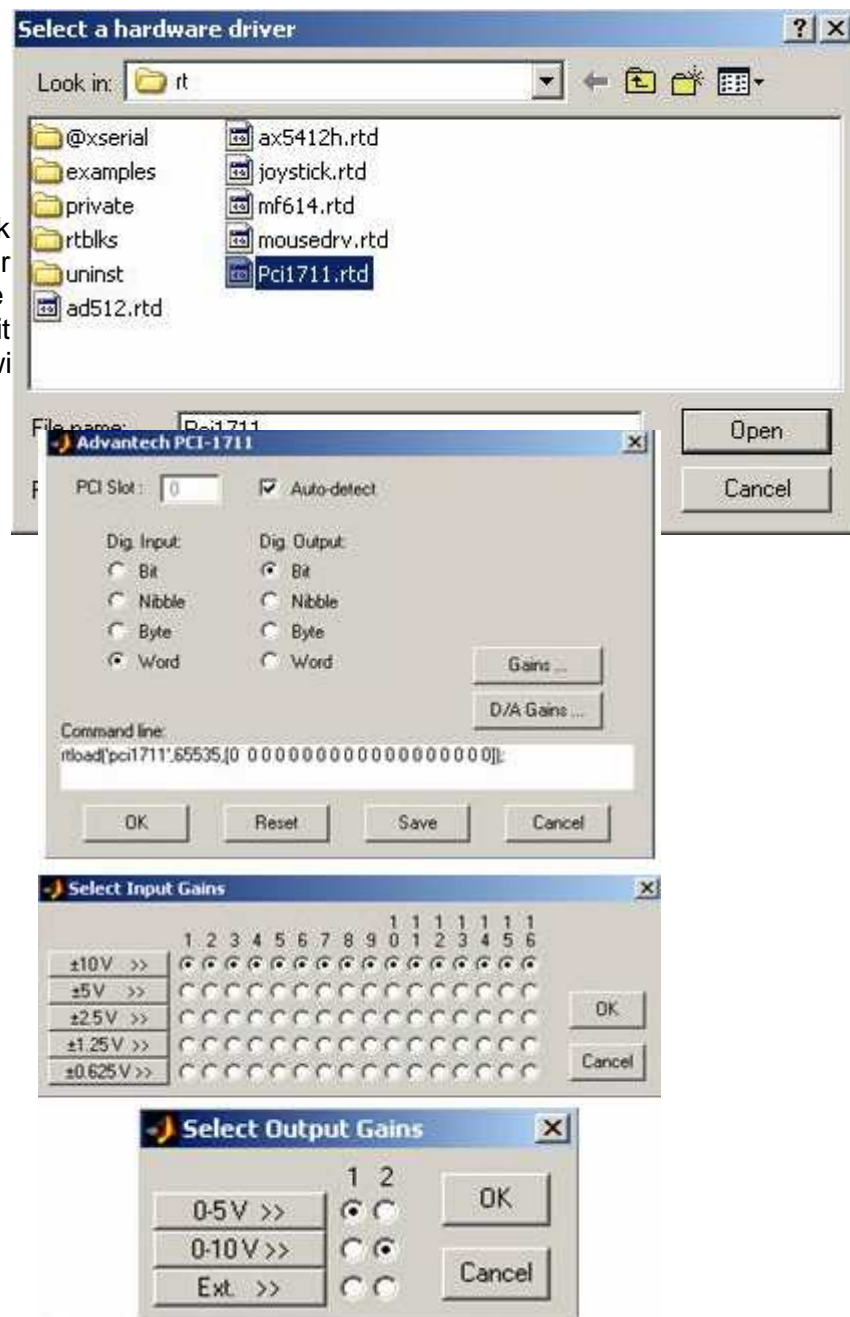


Bild 8.3: Konfiguration des Blocks „Adapter“

Eine Identifikationsmessung an dem realen Objekt ist auf dem Bild 8.4 dargestellt.

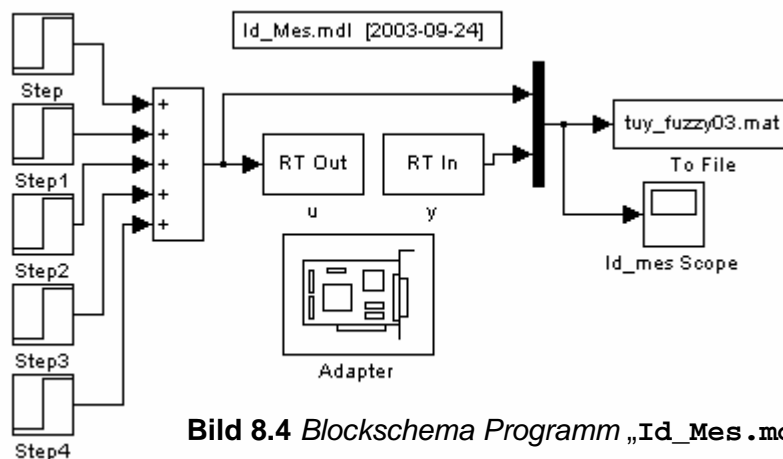


Bild 8.4 Blockschema Programm „Id_Mes.mdl“

Die Untersuchungen der realen Regelstrecke im geschlossenen Regelkreis erfolgen mittels des Programms „PID_Real“. Das Blockschema dieses Programms im SIMULINK ist auf dem Bild 8.5 dargestellt.

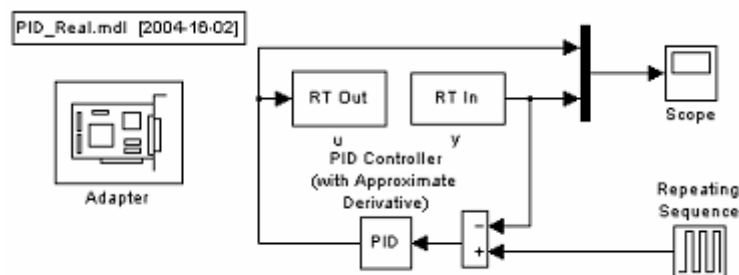


Bild 8.5. Blockschema der realen Regelstrecke mit PID Regler

9. LITERATUR

- [1] GRACE,A.-LAUB,J.A.-LITTLE,J.N.-THOMPSON,C.M.: Control System Toolbox. For Use with MATLAB. User's Guide. The Math Works,Inc.1995
- [2] HOFFMANN, J.: MATLAB und SIMULINK. Beispielorientierte Einführung in die Simulation dynamischer Systeme. Addison-Wesley Longman Verlag, 1998
- [3] MODRLAK, O.: Kurze Einführung in MATLAB
http://www.fm.vslib.cz/~krt/krt_cz/projektovani/Einfuehrung_MATLAB.pdf
- [4] Worlitz,F.: Begleitmaterial zur Vorlesung Projektierung I, Hochschule Zittau/Görlitz, FB Elektro- und Informationstechnik,
- [5] Worlitz,F.: Begleitmaterial zur Vorlesung Projektierung II, Hochschule Zittau/Görlitz, FB Elektro- und Informationstechnik,
- [6] FÖLLINGER,O.: Regelungstechnik. Heidelberg, Hüthig, 1984, 4. Auflage.
- [7] UNBEHAUEN,H.: Regelungstechnik I. Vieweg & Sohn mbH, Braunschweig/Wiesbaden,1992.
- [8] REINISCH, K.: Kybernetische Grundlagen und Beschreibung kontinuierlicher Systeme. VEB Verlag Technik, Berlin,1974