

Für die Durchführung des Versu-  
ches sind die theoretischen Kennt-  
nisse aus dem Kapitel

„Model in the Loop“

der Vorlesung "Entwicklungsprozes-  
se mechatronischer Systeme " er-  
forderlich.

Das Modell soll mit Hilfe von Matlab-  
Simulink aufgebaut und getestet  
werden.

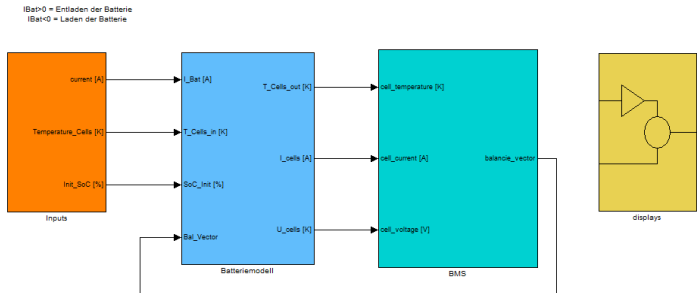


Abbildung 1 MiL-System mit Eingangsgrößen, Batterie-  
modell, BMS und Display

## Einleitung

Ziel des Versuches soll sein, das im Laborversuch erstellte „Batteriemodell“ bestehend aus 12 LiFePo4-Zellen zu nutzen um die Funktion „Limit-Monitoring“ für das Batteriemanagementsys-  
tem zu entwickeln und zu testen.

## Grundlagen

Abbildung 1 zeigt das komplette MiL-System.

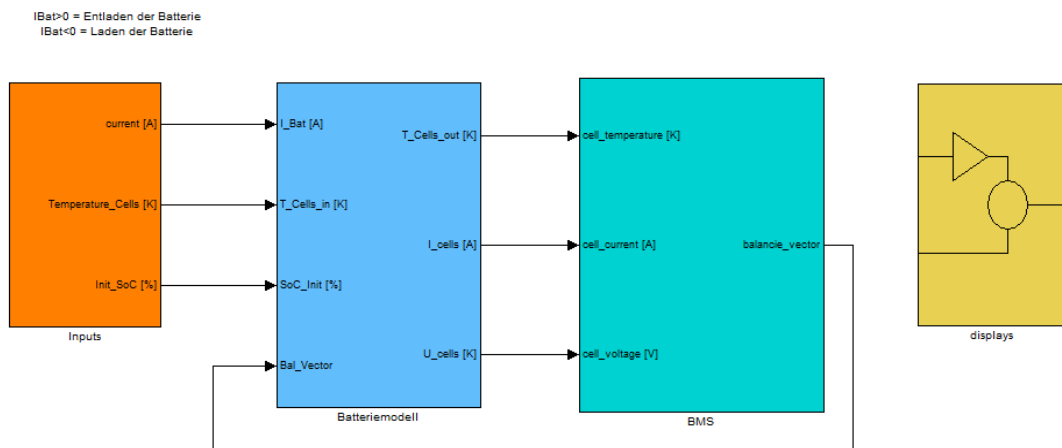
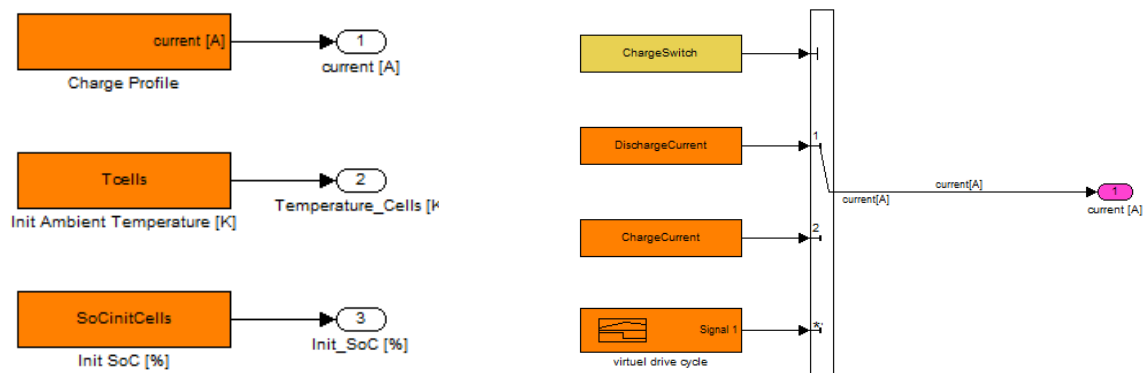


Abbildung 2 MiL-System für ein BMS

Dieses System besteht aus einer Input-Subfunktion (Orange), einem Batteriemodell-Subfunktion (Blau) sowie einem BMS-Funktionsblock (Türkis). Zur Anzeige wird ein Display-Block verwendet, in dem alle relevanten Betriebsdaten des BMS und der Batterie dargestellt werden.

Die im Labor zur Verfügung gestellte Vorlage enthält eine leere Subfunktion für das Batteriemodell sowie eine leere Subfunktion für die Funktionsentwicklung des BMS.

Der Block Inputs enthält alle Startwerte sowie die Möglichkeit, verschiedene Stromprofile einzustellen und mit Hilfe des Batteriemodells die Auswirkung an den Zellklemmen (Klemmspannung) zu simulieren.



**Abbildung 3 Subfunktion Inputs**

Tcell, SoCinitCells, DischargeCurrent, ChargeCurrent sind konstante Vektoren bzw. Konstanten, die in einem M-File vorab (Init\_file\_Bat.m) definiert werden sollten.

In Abbildung 4 wird der Funktionsblock des BMS dargestellt. Hierin enthalten sind Embedded-Matlab Functions, deren Code in den weiteren Laborversuchen entwickelt, getestet und auf dem Steuergerät des BMS-Masters ausgeführt werden soll.

Der eigentliche Matlabcode soll mit Eingangs und Ausgangsgrößen vom Datentyp int16 bzw. uint16 programmiert werden wie dies Abbildung 5 gezeigt ist. Dies ist notwendig, da zu Testzwecken zwischen dem BMS-Master und dem RTPC eine byteweise Signalübertragung über den CAN-Bus erfolgen muss.

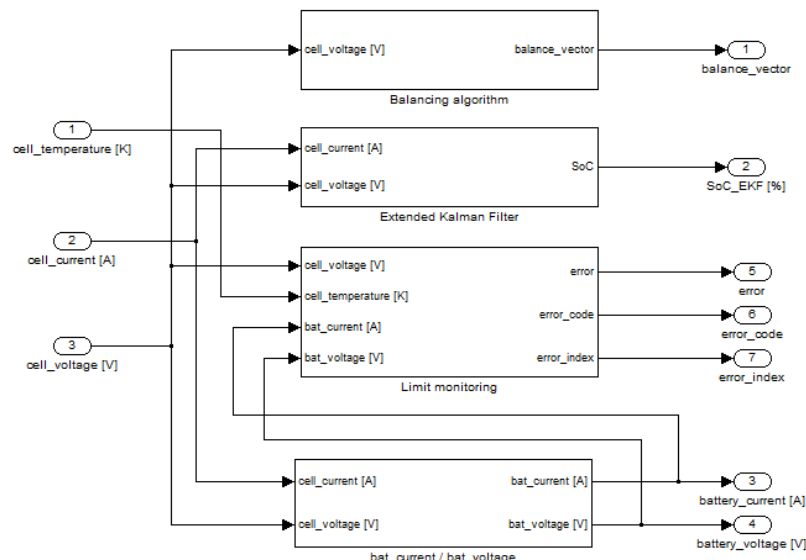


Abbildung 4 Unterfunktionen des Subfunktion-Blocks BMS

Um den Portierungsaufwand für eine spätere Einbettung in den Digitalen Signalprozessor des BMS-Master so gering wie möglich zu halten, wurden die gebrochen rationalen Größen z. B. von V in mV konvertiert. Dies erlaubt eine einfache Umrechnung ohne großen Aufwand.

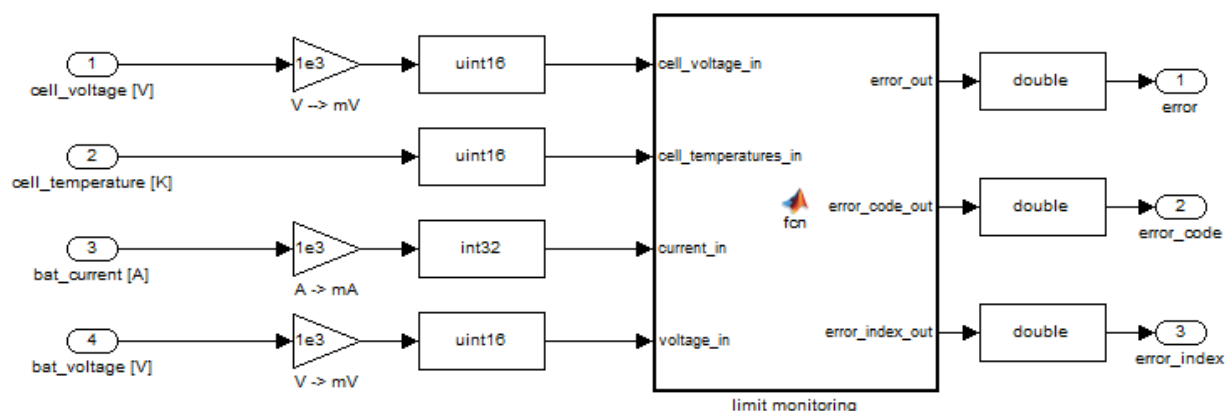
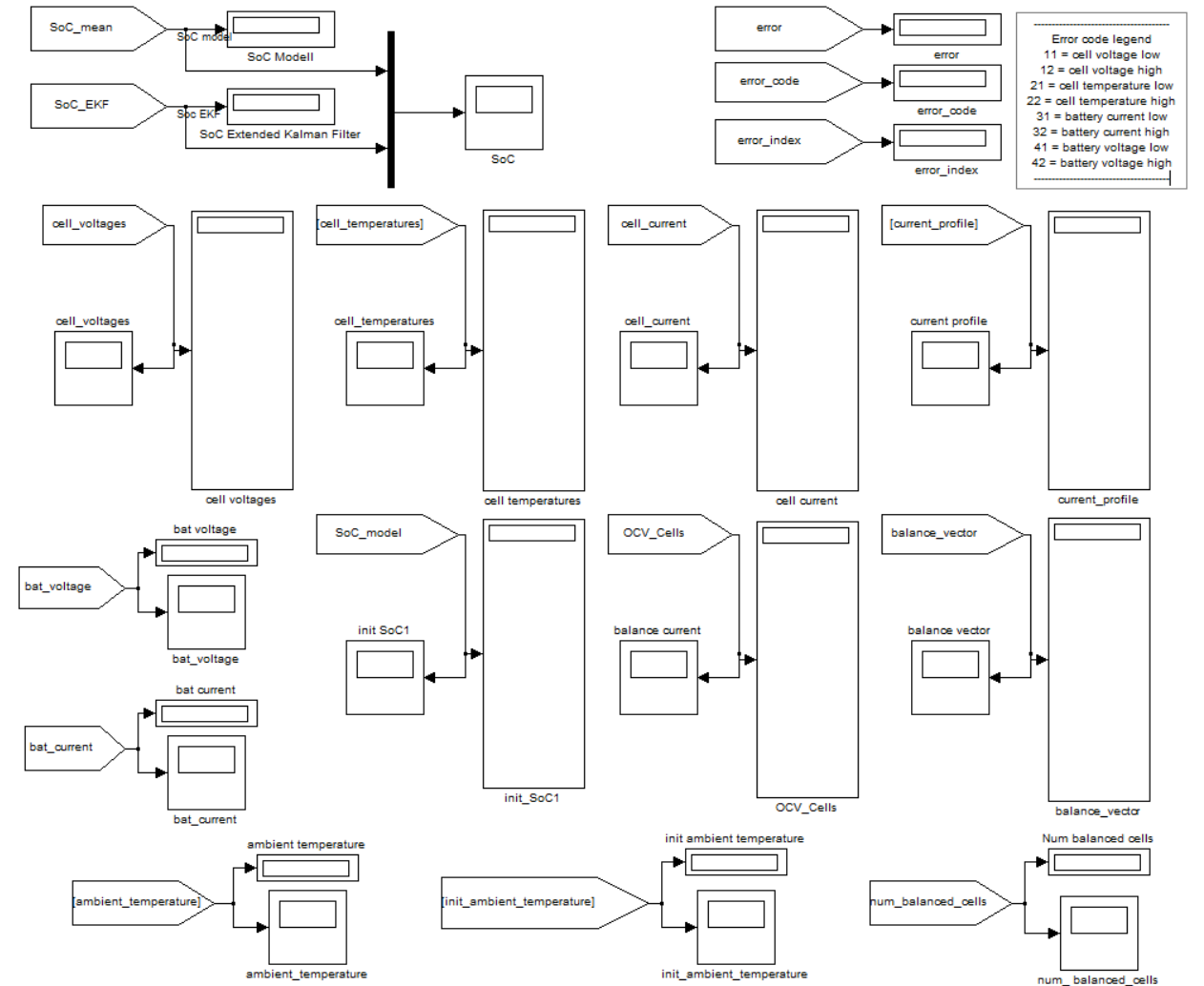


Abbildung 5 Eigentliche Embedded Matlab Funktion des Limit Monitorings

Als Eingangssignale werden alle Zellspannungen, -temperaturen, der Batteriestrom, sowie die gesamte Batteriespannung eingelesen. Als Ausgangssignale sollen drei Error-Meldungen generiert werden, die jeweils ein Byte groß sind (siehe dbf-File aus dem Labor „Kommunikationsmatrix“).

Da die Ergebnisse später im MiL-Versuch weiterverarbeitet werden, werden die Error-Meldungen im Anschluss wieder in Variablen vom Typ `double` konvertiert.



**Abbildung 6 Subfunktion Display mit allen relevanten Anzeigen für die Betriebszustände des Batteriesystems**

Abbildung 6 zeigt den Inhalt der Subfunktion Display. Es bietet es sich an, alle dort definierten Größen aus dem BMS und dem Batteriemodell zur Anzeige zu bringen.

Verwenden Sie zu diesem Zwecke die Signalübertragungsblöcke „From-“ bzw. „Goto-“ Tags aus der Library „Signal-Routing“, um die Signale auf die Subfunktion Display zu übertragen. Achten Sie bei den Einstellungen darauf, dass die Sichtbarkeit dieser Tags auf „global“ eingestellt ist.

## Aufgaben

- 1) Ergänzen Sie die Subfunktion „Batteriemoell“ mit dem Batteriemoell aus dem Laborversuch „Batteriemoellierung“. Fügen Sie in dem Batteriemoell für die gemäß Abbildung 6 dargestellten Anzeigen entsprechende „From“-Blöcke ein. Bearbeiten Sie ggf. das Modell so, dass das eingefügte Batteriemoell die gleichen Ein- und Ausgänge besitzt wie in Abbildung 2 dargestellt.
- 2) Ergänzen Sie ggf. das M-File Init\_file\_Bat.m mit weiteren Parametern.
- 3) Sorgen Sie zunächst dafür, dass der Balancing-Eingang des Batteriemoell mit einem konstanten Vektor bestehend aus 12 Nullen belegt wird (Die Balancing-Funktion soll in einem späteren Versuch ebenfalls entwickelt werden und durch den BMS-Funktionsblock gesteuert werden)
- 4) Das Limit Monitoring soll folgende Grenzwerte überwachen:

### Verfügbare globale Parameter durch das Matlab-File:

```

V_min_cell      = uint16(2800);      % minimal cell voltage [mV]
V_max_cell      = uint16(4090);      % maximum cell voltage [mV]
V_delta_max_cell = unit16(100);      % maximum cell voltage dif-
                                     % ference [mV]

T_min_cell      = uint16(273);      % min. cell temperature [K]
T_max_cell      = uint16(373);      % max. cell temperature [K]
I_max_line      = int32(10000);      % max. charge current [mA]
I_min_line      = int32(-10000);     % max. dischar. current [mA]
V_min_bat       = uint16(36000);     % min. battery voltage [mV]
V_max_bat       = uint16(49080);     % max. battery voltage [mV]
CellNo          = unit8(12)         % numb. of cells in a string
  
```

### Gewünschte Error Codes:

```

code: 00      keine Zelle hat einen Fehler
code: 11      cell voltage low      code: 31      charge current high
code: 12      cell voltage high     code: 32      discharge current high
code: 21      cell temp. low        code: 41      battery voltage low
code: 22      cell temp. high       code: 42      battery voltage high
  
```

### Gewünschter Error Index:

index=zellennummer: Zelle im Strang, die den Fehler aufweist!  
 index=99: Alle Zellen sind betroffen (bei Überschreitung des maximal zulässigen Lade/Entladestroms).

<b>MM118</b> <b>WS</b> <b>14/15</b>	<p>Fachbereich Informatik und Elektrotechnik Grenzstraße 5 24149 Kiel</p>  <p><b>FACHHOCHSCHULE KIEL</b> University of Applied Sciences</p>	<p>Laborversuch (Titel) 03:  <b>Aufbau eines MiL-Systems für die  Funktion Limit Monitoring</b>  Labor für Entwicklungsprozesse mechatroni-  scher Systeme  Prof. Dr.-Ing. Christoph Weber,  Prof. Dr.-Ing. Klaus Lebert</p>	 <p><b>Institut für Mechatronik</b></p>	<p>Seite 6</p>
---	--	--	--	----------------

**Gewünschter Error Event:**

1: Fehler liegt vor.

0: Kein Fehler liegt vor.

- 5) Beschreiben Sie ihre Testfälle, die sie im MiL-Versuchen prüfen. Zeigen Sie dem Laborleiter, wie Sie bei der Prüfung der Testfälle vorgehen.