

SystemC-AMS-Modell eines rückgekoppelten Beschleunigungssensors als Teil eines Inertialnavigationssystems

Markert, Erik; Herrmann, Göran; Müller, Dietmar
TU Chemnitz, Fakultät für Elektrotechnik und Informationstechnik,
Professur Schaltkreis- und Systementwurf, 09107 Chemnitz,
e-mail: erik.markert@etit.tu-chemnitz.de

Kurzfassung

Im Rahmen des SFB 379 entsteht derzeit an der TU Chemnitz ein Inertiales Navigationssystem. Dieses umfasst neben analogen Sensoren und digitaler Hardware auch Softwarekomponenten. In diesem Beitrag wird die Modellierung des Systems mit SystemC-AMS dargestellt. Der Schwerpunkt liegt dabei auf der Nachbildung der Beschleunigungssensoren, die mit Hilfe einer Rückkopplung durch ein PWM-Signal stets im linearen Bereich um die Ruhelage arbeiten. Des weiteren wird ein Überblick über die weiteren Systemkomponenten gegeben.

1 Einleitung

Heutige Navigationssysteme nutzen meist Positionsdaten, die von Satelliten bereit gestellt werden (GPS). Deren Genauigkeit und Verfügbarkeit (Sichtkontakt) ist jedoch vor allem bei der Navigation innerhalb von Gebäuden nicht ausreichend. Daraus ergibt sich die Notwendigkeit, Bewegungen selbst aufzuzeichnen und aus den zurückgelegten Wegen die Position relativ zu einem Anfangspunkt zu ermitteln. Wegen der Messungenauigkeiten der Sensoren ist eine analoge und digitale Fehlerkorrektur unerlässlich.

Kern des vorgestellten inertialen Navigationssystems bilden Beschleunigungs- und Rotationsensoren. Zur Messung von Beschleunigungen kommt eine neuentwickelte Sensorstruktur [1] zum Einsatz. Neben Sensoren und analoger sowie digitaler Fehlerkorrektur umfasst das System auch Softwareanteile. Daher stößt eine Modellierung mit VHDL-AMS, wie in [2] dargestellt, an ihre Grenzen.

Als Alternative wird SystemC-AMS [3] von einer OSCI study group entwickelt. Diese C++-Bibliothek erweitert das digitale SystemC um Methoden zur Modellierung von analogen Problemstellungen. Eine Beta-Version steht auf der SystemC-AMS-Homepage

[3] zum Download zur Verfügung. Die Eignung von SystemC-AMS zur Modellierung von MEMS wurde in [4] und [5] geprüft.

Dieser Beitrag gibt im 2. Abschnitt einen Überblick über das Gesamtsystem. Die Abschnitte 3 und 4 stellen die Modellierung des Beschleunigungssensors dar. In Abschnitt 5 wird das Modell durch Simulation verifiziert. Den Abschluss bildet eine Zusammenfassung.

2 Aufbau des Gesamtsystems

Das Inertialnavigationssystem besteht aus 3 Subsystemen: einem Analogteil mit Fehlerkorrektur, einem Digitalteil mit Koordinatentransformation und einem Softwareteil auf einem PC. Bild 1 gibt einen Überblick über das Gesamtsystem.

Die Sensoren werden durch von der Komponente „environment“ bereitgestellte Stimuli angeregt. Nach der Analog-Digital-Wandlung findet die weitere Bearbeitung im digitalen Teilsystem statt. Neben der digitalen Driftkompensation umfasst der Digitalteil die Koordinatentransformation auf Basis von Quaternionen, die Integration der Beschleunigungen und die Berücksichtigung von Stützstelleninformationen.

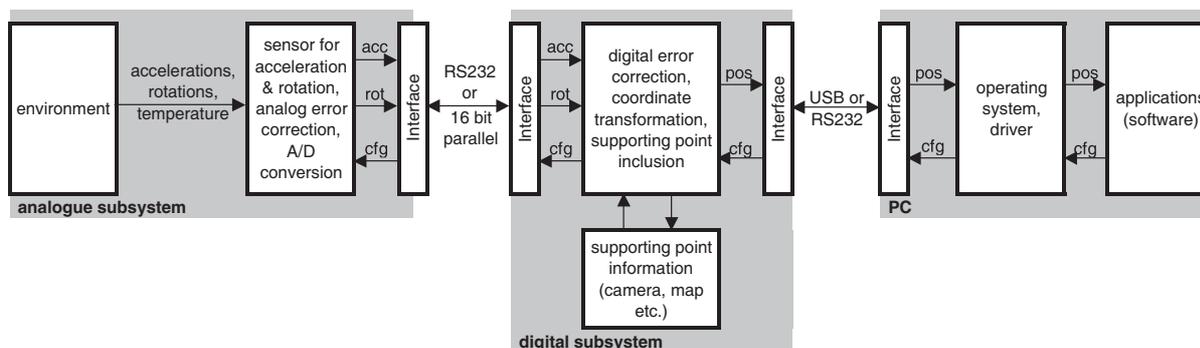


Bild 1: Überblick über das Gesamtsystem

Die so erhaltenen Positionsdaten dienen dem Softwareteil im PC als Grundlage zur Darstellung des zurückgelegten Weges und zur Erzeugung von Steuerinformationen. Dieses Subsystem unterteilt sich in zwei Komponenten: Das als SystemC-Modul ausgeführte Betriebssystem und die Applikationen, welche als Threads auf dem Betriebssystem aufsetzen.

Die Schnittstellen zwischen den Subsystemen werden durch SVE-Interfaces [6] repräsentiert. SVE bezeichnet eine SystemC-Erweiterungsbibliothek zur abstrakten Modellierung von Datentransfers. Damit ist es möglich, verschiedene Protokolle zur Datenübertragung zu testen und für die Festlegung des zu nutzenden Übertragungsmodus' Vergleichssimulationen durchzuführen.

Das Modell des Beschleunigungssensors hat einen großen Einfluss auf die Genauigkeit des Gesamtsystemmodells. Daher soll es im Abschnitt 3 im Mittelpunkt der Betrachtungen stehen. Die übrigen Komponenten werden in den folgenden Teilabschnitten kurz erläutert.

2.1 Analoges Subsystem

Das analoge Subsystem besteht aus sechs Sensoren, analoger Fehlerkorrektur und Analog-Digital-Wandlern. Von den sechs Sensoren messen drei die einwirkenden Beschleunigungen und drei die Rotationen an den drei Achsen. Die analoge Fehlerkorrektur befindet sich derzeit noch in der Entwicklungsphase. Die Analog-Digital-Wandler für die Beschleunigungssignale besitzen eine Auflösung von 12 bit bei einer maximalen Umsetzrate von 1,2 MSample/s. Bild 2 zeigt die Teilkomponenten im Analogbereich.

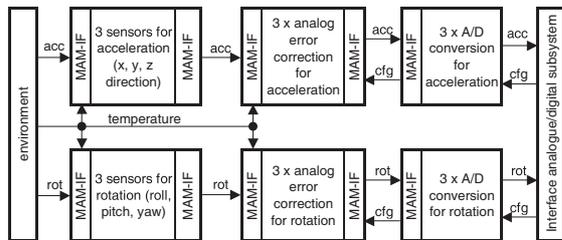


Bild 2: Analoges Teilsystem

Um einen einfachen Wechsel der Abstraktionsebenen in den Teilkomponenten zu ermöglichen, sind die Schnittstellen als MAM-Interfaces implementiert. Die Multi-Architecture Modellierung (MAM) [7] verbindet alle Komponenten auf niedrigster Abstraktions-

ebene. Die MAM-Schnittstellen konvertieren die höherabstrakten Ausgangssignale in elektrische Spannungen und nehmen an den Komponenteneingängen die umgekehrte Operation vor. Bei SystemC-AMS stehen neben den digitalen SystemC-Signalen mit konservativer und Signalflussebene zwei Abstraktionsebenen zur Verfügung, die im Kapitel 4 näher erläutert werden. Bei einem Anschluss einer Komponente auf konservativer Ebene (ELEC) an eine Komponente auf Signalflussebene (SDF) ergibt sich eine Schnittstellenbeschaltung wie in Bild 3 dargestellt.

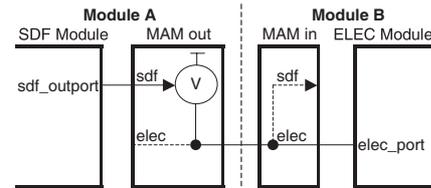


Bild 3: Multi-Architecture Schnittstelle

2.2 Digitales Subsystem

Die gewandelten Analogsignale der Sensoren benötigen eine weitere Aufbereitung. Im digitalen Subsystem soll dazu eine Korrektur von Drift- und Temperatureinflüssen erfolgen. Des Weiteren müssen die gemessenen Rotations- und Beschleunigungsdaten vom Koordinatensystem der Sensoren in das globale Koordinatensystem gedreht werden. Dies geschieht auf Basis von Quaternionen, komplexen Zahlen mit drei Imaginäranteilen [8]. Im Gegensatz zum Verfahren mit Eulerschen Winkeln kann hiermit die Rotation um alle drei Achsen gleichzeitig berechnet werden.

Für die Integration der Messwerte kommt die Simpson-Regel zum Einsatz, die mit wenig Hardwareaufwand eine genaue Berechnung ermöglicht. Hierbei wird die zu integrierende Funktion durch Funktionen zweiten Grades angenähert.

Die weitere Fehlerkorrektur erfolgt mittels Kalman-Filterung auf Basis von Stützstelleninformationen. Diese werden von einer Karte bereitgestellt.

Zur Realisierung dieses Subsystems stehen im Prototyp vier Xilinx VirtexE FPGAs zur Verfügung. Bild 4 gibt einen Überblick über das digitale Subsystem und dessen Partitionierung auf die vier FPGAs.

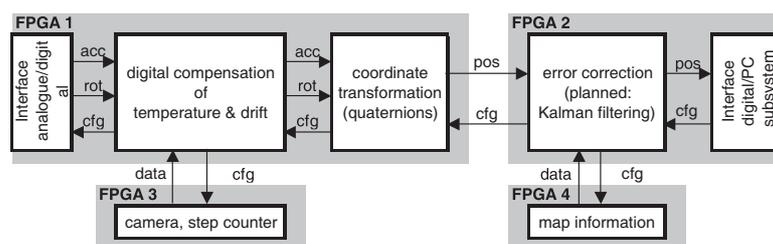


Bild 4: Digitales Teilsystem

3 Modellierung des rückgekoppelten Beschleunigungssensors

Der Sensor besteht aus einer Masse- und vier Kammsegmenten zur kapazitiven Beschleunigungsdetektion. Bild 5 zeigt die Grundstruktur des Sensors. Eine ein-

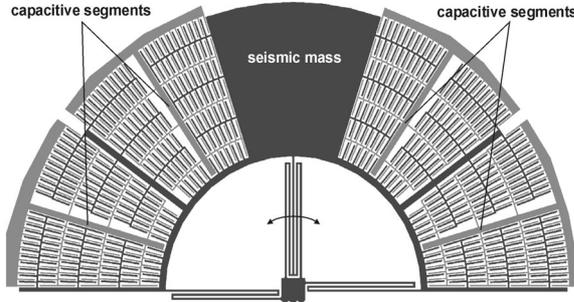


Bild 5: Struktur des Sensors

wirkende Beschleunigung führt zu einer Verschiebung der Masse. Die drei Federn zwingen die Struktur auf eine Kreisbahn. Die Kammstrukturen sind jeweils gegeneinander ausgerichtet, d.h. eine Kapazitätserhöhung in einem Segment führt im Nachbarsegment zu einer Verringerung der Kapazität. Zur Messung werden an die Kämme Spannungen angelegt. Aus dem nach dem Differentialkondensatorprinzip fließenden Strom (siehe Bild 7) kann die einwirkende Beschleunigung bestimmt werden. Durch unterschiedliche Positionen der Masse lassen sich Sensoren mit verschiedenen Detektionsachsen realisieren.

Zur Vereinfachung der Berechnung des Verhaltens wird die Struktur, wie in Bild 6 zu sehen, in 11 Punkte zusammengefasst. Dabei sind die 5 Segmente auf ihre

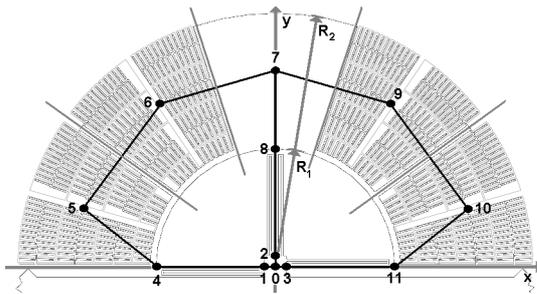


Bild 6: Vereinfachung der Sensorstruktur

Massenschwerpunkte reduziert. Das Verhalten zwischen ihnen wird durch flexible Verbindungen dargestellt. Außerdem wird das System in ein mechanisches (Auslenkung), ein elektrisches (Kapazitätsberechnung) sowie ein Auswerte- und Rückkopplungsteilsystem untergliedert. Die Herleitung der Modellgleichungen für das mechanische und das elektrische Teilsystem ist in [2] beschrieben. An dieser Stelle sollen lediglich die Ergebnisse genannt werden.

3.1 Mechanisches Teilsystem

Die rotatorische Beschleunigung eines Segments a_s ist abhängig von den translatorischen Beschleunigungen a_x und a_y , der momentanen Auslenkung α und der Position des Segments β_s :

$$a_s = a_x \cdot \sin(\alpha + \beta_s) + a_y \cdot \cos(\alpha + \beta_s) \quad (1)$$

Daraus ergibt sich das auf das Segment wirkende Drehmoment M_s in Abhängigkeit der Masse des Segments m_s sowie der Segmentschwerpunktkoordinaten x_s und y_s :

$$M_s = m_s \cdot a_s \cdot \sqrt{x_s^2 + y_s^2} \quad (2)$$

Die Bewegungsgleichung für Rotationen präzisiert sich damit wie folgt:

$$\sum M_s - \sum M_R = \sum J_s \cdot \frac{d^2 \alpha}{dt^2} + k \cdot \frac{d\alpha}{dt} + \sum c_{rot} \cdot \alpha \quad (3)$$

Dabei gibt M_R das durch die angelegte Spannung verursachte Rückstellmoment eines Segments an. Das Trägheitsmoment J_s und die Federkonstante c_{rot} berechnen sich aus Geometriedaten und Materialkonstanten. Durch Lösung der Differentialgleichung erhält man die Auslenkung α des Sensors.

3.2 Elektrisches Teilsystem

Aus der im mechanischen Teilsystem bestimmten Auslenkung α resultiert eine Änderung der Kapazitäten der Segmente C_s . Zusammen mit der angelegten Messspannung U_s bewirkt dies einen Stromfluss I_s :

$$I_s = U_s \cdot \frac{dC_s}{dt} + C_s \cdot \frac{dU_s}{dt} \quad (4)$$

Außerdem verursachen Messspannung und Kapazitätsänderung ein Rückstellmoment M_R im mechanischen Teilsystem:

$$M_R = \frac{1}{2} \cdot U_s^2 \cdot \sqrt{x_s^2 + y_s^2} \cdot \left(\frac{dC_{s1}}{ds} + \frac{dC_{s2}}{ds} \right) \quad (5)$$

3.3 Auswerte- und Rückkopplungsteilsystem

Die Abhängigkeit des Rückstellmoments und damit der Auslenkung von der angelegten Spannung wird zum Rückführen des Sensors in die Ausgangsposition genutzt. Dazu werden die Segmentspannungen pulswellenmoduliert. Bild 7 zeigt den prinzipiellen Aufbau der Auswerte- und Rückkopplungsschaltung.

Die aus einer Beschleunigung resultierende Kapazitätsänderung bewirkt einen Stromfluss am aus 2 Sensorsegmenten gebildeten Differentialkondensator. Dieser Stromfluss wird mittels integrierendem OPV in eine Spannung gewandelt, die sich proportional zur anliegenden Beschleunigung verhält. Ein Oszillator nutzt die Polarität der Spannung zur Erstellung von pulswei-

tenmodulierten Spannungen, welche an den Segmenten Rückstellmomente erzeugen.

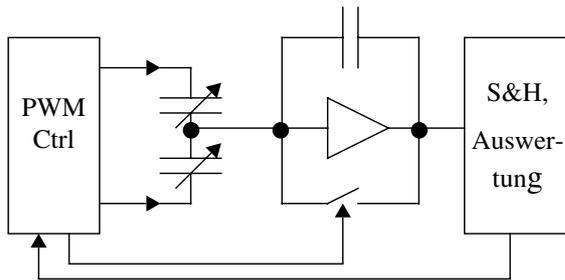


Bild 7: Auswerte- und Rückkopplungsschaltung

4 Implementierung des Sensors mit SystemC-AMS

4.1 Modellierungsmöglichkeiten mit SystemC-AMS

Zur Beschreibung von analogen Systemanteilen stehen in der genutzten SystemC-AMS-Version konservative und Datenfluss-Netze (SDF, static dataflow) zur Verfügung. In SDF-Netzen kommunizieren die Einzelelemente über gerichtete Datenströme miteinander. Jede Schnittstelle eines Moduls ist als Ein- oder Ausgang zu charakterisieren. Rückführungen im Signalfluss sind erlaubt, jedoch müssen sie in der aktuellen Version durch das Einfügen eines Verzögerungsgliedes entkoppelt werden. In Kommunikationssystemen wird von einer starken Überabtastung der Signale ausgegangen, so dass in diesem Anwendungsgebiet die Verzögerung zu keinem systemrelevanten Fehler führt. Für die Beschreibung von Transferfunktionen und Differentialgleichungen steht eine Funktion zur Lösung von Laplace-Übertragungsfunktionen zur Verfügung.

Neben der Modellierung von Datenflussnetzen bietet SystemC-AMS die Möglichkeit der Beschreibung linearer elektrischer Netze, die per MNA (Modified Nodal Analysis) gelöst werden. Dazu können verschiedene Bauelemente genutzt werden: Widerstände, Spulen, Kondensatoren, spannungsgesteuerte Strom- und Spannungsquellen sowie stromgesteuerte Spannungs- und Stromquellen. Zur Anbindung an analoge und digitale Datenfluss-Netze stehen SDF- bzw. SystemC-signalgesteuerte Strom- und Spannungsquellen sowie Schalter bereit. Als Rückwandler fungieren Ampere- und Voltmeter, deren „Messwert“ als SDF-Signal abgegriffen werden kann.

Mischformen mit digitalem SystemC 2.0 und hierarchische Anordnungen sind ebenfalls gestattet.

4.2 Sensormodell

Die Nutzung von SDF-Netzen beschleunigt die Simulation in SystemC-AMS gegenüber einer Implementierung mit konservativen Netzen und minimiert den Beschreibungsaufwand. Das vorgestellte Sensormodell stützt sich daher überwiegend auf diesen gerichteten

ten Netztyp. Bedingt durch die notwendige Entkopplung der Rückführungen ergibt sich die in Bild 8 dargestellte Struktur. Um den Einfluss der Verzögerungen

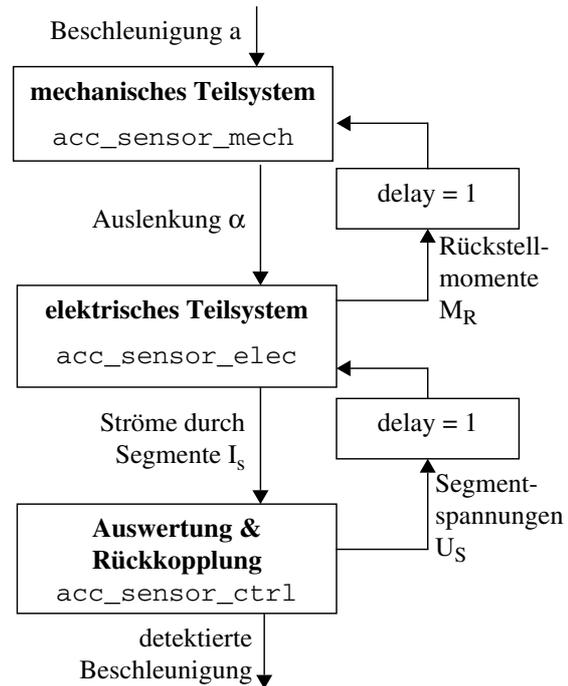


Bild 8: Struktur des SDF-Modells des Sensors

rungsglieder auf das Systemmodell zu prüfen, wurde ein Vergleichsmodell in VHDL-AMS erstellt. Dabei zeigte sich keine relevante Abweichung im Systemverhalten. Listing 1 zeigt die Umsetzung der Systemstruktur aus Bild 8 in SystemC-AMS.

```
#include "systemc-ams.h"
#include "acc_sensor_ctrl.h"
#include "acc_sensor_mech.h"
#include "acc_sensor_elec.h"

class acc_sensor_single : public
sc_module{
    sca_sdf_in<double> ax, ay;
    sca_sdf_in<double> temp;
    sca_sdf_out<double> a_sensor;

    SC_HAS_PROCESS(acc_sensor_single);

    acc_sensor_elec *sel1;
    acc_sensor_ctrl *sst1;
    acc_sensor_mech *sm1;
    sca_sdf_signal<double> alphas, mk1,
        mk2, mk3, mk4, mx;
    sca_sdf_signal<double> u1, u2, u3,
        u4, i1, i2, i3, i4, C1, C2, C3, C4;

    acc_sensor_single(sc_module_name
name, int pos_mass):sc_module(name){
        sm1 = new
```

```

acc_sensor_mech("SM1",pos_mass);
sm1->ax(ax); sm1->ay(ay);
sm1->alpha(alpham); //Ausgang
sm1->Mk1(mk1);
sm1->Mk1.set_delay(1);
sm1->Mk2(mk2);
sm1->Mk2.set_delay(1);
sm1->Mk3(mk3);
sm1->Mk3.set_delay(1);
sm1->Mk4(mk4);
sm1->Mk4.set_delay(1);

sell = new
  acc_sensor_elec("SEL1");
sell->u1(u1);
sell->u1.set_delay(1);
sell->u2(u2);
sell->u2.set_delay(1);
sell->u3(u3);
sell->u3.set_delay(1);
sell->u4(u4);
sell->u4.set_delay(1);
sell->alpha(alpham);
sell->i1(i1); sell->i2(i2);
sell->i3(i3); sell->i4(i4);
sell->C1(C1); sell->C2(C2);
sell->C3(C3); sell->C4(C4);
sell->Mk1(mk1); sell->Mk2(mk2);
sell->Mk3(mk3); sell->Mk4(mk4);

sst1 = new
  acc_sensor_ctrl("SC1");
sst1->i1(i1); sst1->i2(i2);
sst1->i3(i3); sst1->i4(i4);
sst1->C1(C1); sst1->C2(C2);
sst1->C3(C3); sst1->C4(C4);
sst1->u1(u1); sst1->u2(u2);
sst1->u3(u3); sst1->u4(u4);
sst1->temp(temp);
sst1->a_sensor(a_sensor);
}};

```

Listing 1: Top-Level des Einzelsensors in SystemC-AMS

Innerhalb des elektrischen und mechanischen Teilsystems wird das Verhalten unter Nutzung von Laplace-Transferfunktionen basierend auf den Gleichungen aus Abschnitt 3 nachgebildet. Im Auswerte- und Rückkopplungsmodul kommt neben SDF-Strukturen zur PWM-Generierung ein elektrisches Netz zur Nachbildung des OPV mit seiner Außenbeschaltung (siehe Bild 7) zum Einsatz. Listing 2 zeigt den zugehörigen SystemC-AMS-Code.

```

#include "systemc-ams.h"

SC_MODULE(I_U_conversion){
  sca_sdf_in<double> i_in,switch_on;
  sca_sdf_out<double> v_out;

```

```

  sca_r *rcap; sca_c *cap;
  sca_sdf2r *rsw; sca_sdf2i *isrc;
  sca_vcvs * opv; sca_v2sdf *vsink;
  sca_elec_wire opv_in, opv_out,
    back1;
  sca_elec_ref gnd;

  SC_CTOR(I_U_conversion){
    rcap = new sca_r("r1",1);
    rcap->p(opv_out); rcap->n(back1);

    cap = new sca_c("c1",10e-12);
    cap->p(back1); cap->n(opv_in);

    isrc = new sca_sdf2i("Isrc1",1);
    isrc->p(opv_in); isrc->n(gnd);
    isrc->ctrl(i_in);

    opv = new sca_vcvs("OPV1",1e5);
    opv->np(opv_out); opv->nn(gnd);
    opv->ncp(opv_in); opv->ncn(gnd);

    vsink= new sca_v2sdf("VSINK");
    vsink->p(opv_out);
    vsink->sdf_voltage(v_out);

    rsw = new sca_sdf2r("RSW1",10e6);
    rsw->p(opv_out); rsw->n(opv_in);
    rsw->ctrl(switch_on);
  }
};

```

Listing 2: Realisierung der OPV-Schaltung in SystemC-AMS

5 Simulationsergebnisse

Zur Verifikation der Modelle des mechanischen und des elektrischen Teilsystems wurde in einer ersten Simulation die Rückkopplungsschaltung deaktiviert und das Verhalten mit der in [2] beschriebenen VHDL-AMS-Implementierung verglichen. Dabei traten im dynamischen Fall bei Anlegen einer Sprungfunktion geringe Abweichungen von ca. 2% in der Amplitudenhöhe des Ausgangssignals auf.

Auswertungs- und PWM-Parameter sind derzeit noch Gegenstand der Forschung. Bild 9 zeigt die Auslenkung α der Sensorstruktur bei einer PWM-Periode von 14 kHz und einem Spannungshub von 1 V als Reaktion auf eine sprungförmige Beschleunigungsänderung. Darin wird deutlich, dass die Auslenkung durch die PWM-Rückkopplung fast auf den Ruhezustand zurückgeführt wird

Die Auswerteeinheit berechnet aus der aktuell eingestellten Pulsweite die Beschleunigung als Ausgangswert für die weiteren in Abschnitt 2 aufgeführten Subsysteme.

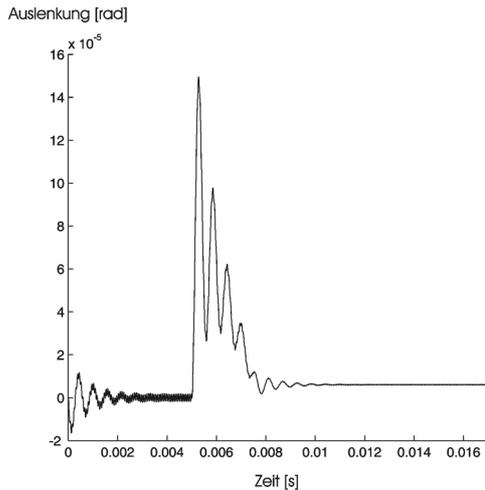


Bild 9: Sprungantwort mit PWM-Rückkopplung

6 Zusammenfassung

Der Beitrag stellt die Modellierung eines heterogenen Inertialnavigationssystems vor. Der Schwerpunkt liegt dabei auf der Beschreibung des analogen rückgekoppelten Beschleunigungssensors. Dieser wurde mit SystemC-AMS modelliert. Zur Teilverifikation des entstandenen Modells diente ein bereits vorhandenes VHDL-AMS Modell. Die zusätzlich implementierte PWM-Rückkopplungsschaltung befindet sich noch in der Entwicklungsphase, die verwendeten Parameter werden derzeit in Zusammenarbeit mit dem Teilprojekt „Komponentenentwurf“ überprüft.

7 Dank

Die hier vorgestellte Arbeit entstand im Rahmen des Teilprojektes A2 „Systementwurf“ in Zusammenarbeit mit dem Teilprojekt A1 „Komponentenentwurf“ des von der DFG geförderten SFB 379 „Mikromechanische Sensor- und Aktorarrays“.

Ein spezieller Dank gilt Herrn Karsten Einwich (FhG Dresden) für die Bereitstellung der SystemC-AMS-Bibliothek, Version 0.13beta1.

8 Literatur

- [1] Billep, D., Dienel, M., *Patentanmeldung DE 102004046411.1 Beschleunigungssensor*
- [2] Markert, E.; Schlegel, M.; Dienel, M.; Herrmann, G.; Müller, D.: *Modeling of a new acceleration sensor as part of a 2D sensor array in VHDL-AMS*. Nanotech 2005, Anaheim CA, Mai 2005
- [3] Einwich, K. et. al.: *White Paper SystemC-AMS Study Group*. <http://www.systemc-ams.org>
- [4] Markert, E.; Schlegel, M.; Herrmann, G.; Müller, D.: *Beschreibung von mechatronischen Systemen mit SystemC-AMS*. 10. GMM-Workshop „Methoden und Werkzeuge für den Entwurf von Mikrosystemen“, Cottbus, Oktober 2004
- [5] Markert, E.; Schlegel, M.; Michel, M.; Herrmann, G.; Müller, D.: *Untersuchung der Anwendbarkeit von SystemC-AMS bei der Beschreibung von MEMS*. 5. GMM/GI/ITG-Workshop „Multi-Nature Systems“, Dresden, 18. Februar 2005
- [6] Siegmund, R.; Müller, D.: *SystemC^{SV}-Extension of SystemC for Mixed Multi-Level Communication Modeling and Interface-based System Design*. DATE 2001, München, März 2001, ISBN 0-7695-0993-2
- [7] Schlegel, M.; Herrmann, G.; Müller, D.: *Application of the 'Multi Architecture Modelling' design method to system level MEMS simulation*. DTIP 2003, Mandelieu - La Napoule; Mai 2003.
- [8] Hart, J. C.; Francis, G. K.; Kauffman, L. H.: *Visualizing Quaternion Rotation*. ACM Transactions on Graphics, Vol.13, No. 3, pp 256-276, Juli 1994.