

Untersuchung der Anwendbarkeit von SystemC-AMS bei der Beschreibung von MEMS

Markert, Erik; Schlegel, Michael; Michel, Matteo; Herrmann, Göran; Müller, Dietmar
TU Chemnitz, Fakultät für Elektrotechnik und Informationstechnik,
Professur Schaltungs- und Systementwurf, Reichenhainer Str. 70, 09126 Chemnitz
e-mail: erik.markert@etit.tu-chemnitz.de

Abstract

Zur Beschreibung von Systemen mit Software- sowie analogen und digitalen Hardwarekomponenten ist an der Außenstelle Entwurfsautomatisierung des Fraunhofer-Instituts für Integrierte Schaltungen Dresden mit SystemC-AMS eine neue Modellierungsmöglichkeit geschaffen worden. Anwendungsgebiet der vorliegenden ersten Version sind Kommunikationssysteme. In diesem Beitrag wird untersucht, ob sich diese Version der Erweiterungsbibliothek bereits für die Modellierung mikromechanischer Komponenten eignet.

1 Einleitung

Mit SystemC [3] ist ein umfangreiches Werkzeug für den Entwurf von digitalen Hardware/Software-Systemen von der funktionellen bis hin zur Register-Transfer-Ebene vorhanden. Heutige Systeme weisen zumeist eine heterogene Struktur auf, d. h. sie bilden eine Gesamtheit aus analogem und digitalem Verhalten in Verbindung mit Softwarekomponenten. Daher ist sowohl eine Erweiterung der Beschreibungssprache als auch der Berechnungsmodelle erforderlich. Des Weiteren verlangen analoge Komponenten detailliertere Modelle als digitale Komponenten. Aus diesem Grund entwickelt eine SystemC Study Group die SystemC-Erweiterungsbibliothek SystemC-AMS. Die in Phase 1 des White Papers [1] vorgesehenen Sprachelemente stehen nun in der Version 0.12 der FhG EAS IIS Dresden bereit. Die umgesetzten Algorithmen sind in [2] erläutert.

Mit dieser Bibliothek soll ein im Rahmen des Sonderforschungsbereiches 379 an der TU Chemnitz entwickelter Demonstrator mit kapazitivem Vibrationssensor beschrieben werden. Das System umfasst einen Sensor sowie analoge und digitale

Signalverarbeitung. Damit eignet es sich besonders gut für die Untersuchung der Beschreibung heterogener Systeme. In Bild 1 wird der grundlegende Systemaufbau dargestellt.

Das Sensor-Array besteht aus einer Vielzahl lateral beweglicher Feder-Masse-Systeme, welche frequenzselektiv in Resonanz arbeiten. Die Auswahl eines einzelnen Sensors obliegt dabei dem Steuerteil des Gesamtsystems. Die am Eingang aufgenommenen Werte werden dem Sensor-Array zugeführt und an dessen Ausgang einer analogen Signalverarbeitung bereitgestellt. Diese Daten dienen als Grundlage für die digitale Steuerung. In der Klassifikationskomponente erfolgt die Einteilung der im Steuerteil gewandelten Signale mittels Fuzzy-Pattern-Algorithmen in Güteklassen [4].

Als Anwendungsbereich werden Maschinen gesehen. Durch das Aufbringen der Sensoren auf verschleißkritische Maschinenteile können Vibrationen klassifiziert werden und somit eine frühzeitige Erkennung der Abnutzung erfolgen. Bild 2 zeigt ein Foto des Sensors.

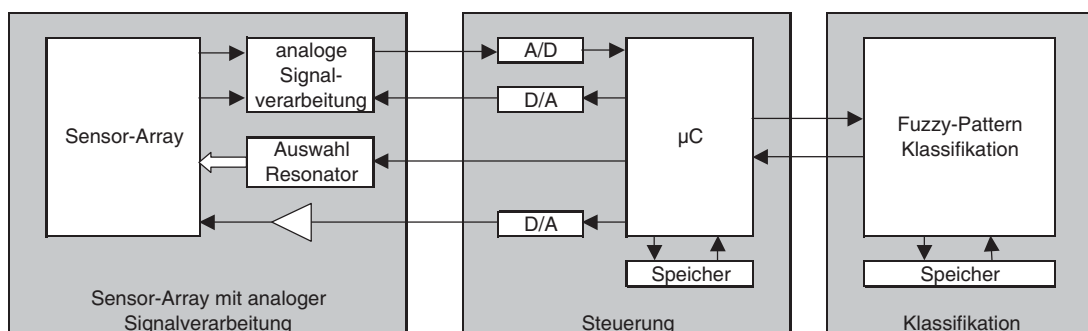


Bild 1: Systemaufbau des Demonstrators zum Vibrationssensor-Arrays [5]

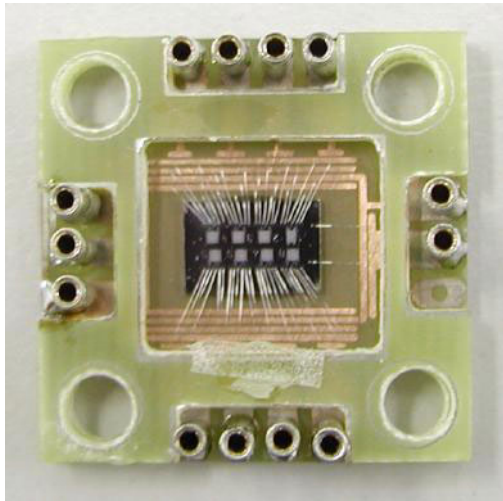


Bild 2: Foto des Vibrationssensorarrays [7]

2 Aufbau des Systems

Das Demonstratorsystem ist in die drei Funktionsbereiche Steuerung, Messung und die Verarbeitung der Signale mit anschließender Klassifikation eingeteilt. Die Funktionen werden wie in Bild 3 dargestellt von den 5 Untermodulen μc , fpc, selector, detector und sensor umgesetzt.

Der Mikrocontroller (μc) steuert die Abläufe innerhalb des Systems und führt die notwendigen Digital-Analog- und Analog-Digital-Wandlungen durch. Er ist mit der rein digitalen Komponente Fuzzy-Pattern-Klassifikator (fpc) verbunden, welche den Klassifizierungsprozess umfasst. Der Selector wählt aus dem Array von 8 Sensoren die Subkomponente mit passender Resonanzfrequenz aus. Zur Feinabstimmung der Sensoren dient die vom Mikrocontroller bereitgestellte Tuning-Span-

nung (v_tun). Das Modul detector enthält die analoge Signalverarbeitung. Diese umfasst einen Lock-In-Verstärker sowie Komponenten zur Tiefpassfilterung, Summation und Verstärkung.

3 Modellierung 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 Einzelemente 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, welche per MNA (Modified Nodal Analysis) gelöst werden. Dazu stehen 7 Bauelemente zur Verfügung: R, L, C, spannungsgesteuerte Strom- und Spannungsquellen sowie stromgesteuerte Spannungs- und Stromquellen. Zur Anbindung an Datenfluss-Netze stehen SDF-gesteuerte Strom- und Spannungsquellen sowie Widerstände bereit. Als Rückwandler fungieren elec2sdf-Signale, also konservative

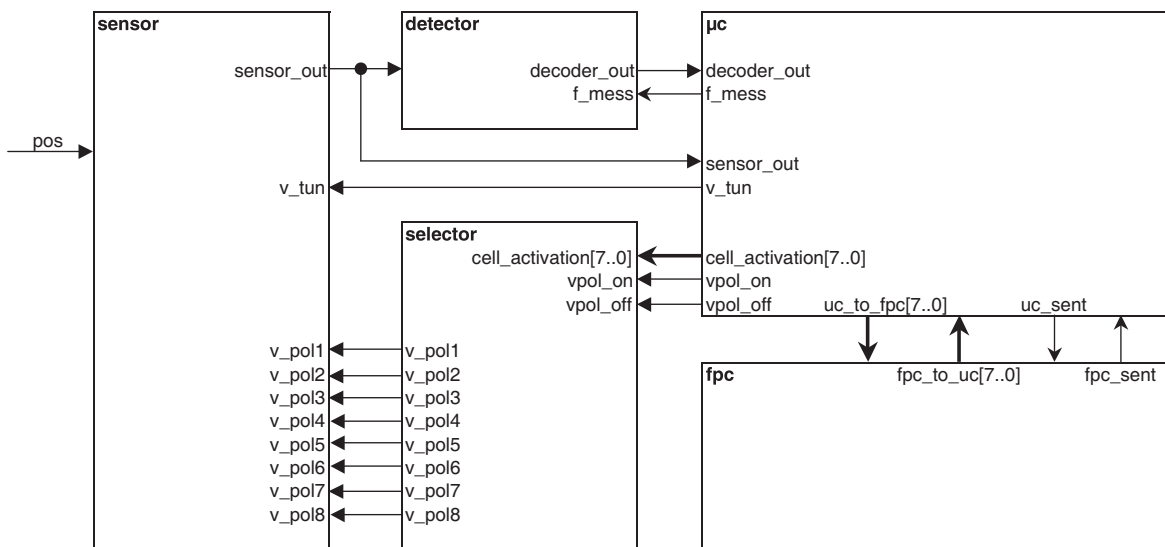
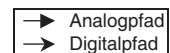


Bild 3: Systemaufbau in SystemC-AMS [5]



Knoten, deren Potentialwert als SDF-Signal abgegriffen werden kann.

Mischformen mit digitalem SystemC 2.0 und hierarchische Anordnungen sind durch Implementierung spezieller Konstruktoren ebenfalls gestattet.

Die verschiedenen Modellierungsmöglichkeiten werden in [6] anhand eines Beispiels näher erläutert.

4 Implementierung

Gesamtsystem

Da SystemC-AMS eine Erweiterung von SystemC und damit C++ ist, bietet es sich an, rein digitale Komponenten mit SystemC zu beschreiben. Dies hat den Vorteil, dass flankengesteuerte Prozesse (SC_THREADS bzw. SC_METHODS) eingesetzt werden können, die im Vergleich zu analogen Anweisungen weniger Rechenzeit beanspruchen und weniger Beschreibungsaufwand (vor allem bei sequentiellen Abläufen) erfordern. Im Vibrationssensorsystem wird daher der Klassifikator in digitalem SystemC beschrieben. Für Mikrocontroller und Detektor kommen hierarchische analog-digitale Module zum Einsatz, während Sensor und Selector rein analoge SystemC-AMS-Komponenten sind. Die Einzelkomponenten werden in [5] näher erläutert. Im folgenden stehen mit den Modulen Sensor und Fuzzy-Pattern-Klassifikator je eine analoge und eine digitale Komponente im Fokus der Betrachtung.

Modul Sensor

Das Sensor-Modul hat die Funktion, am Eingang anliegende, mechanische Signale in elektrische Signale zu wandeln. Zu diesem Zweck ist eine Anzahl lateral beweglicher Feder-Masse-Resonatoren notwendig, die jeweils einzeln aktiviert werden können. Die Ausgangsschnittstelle `u_out` vom SDF-Typ `double` stellt eine elektrische Spannung für das Nachfolgemodul zur Verfügung.

Mit SystemC-AMS ist es in der vorliegenden Version 0.12 im Gegensatz zu VHDL-AMS nicht möglich, konservative Module mechanischer Sachverhalte direkt mittels Differentialgleichungen, Fluss- und Differenzgrößen zu entwerfen. Dieses ist in einer der nächsten Versionen vorgesehen. Ein mögliches Prinzip wird in [9] demonstriert. Lineares Verhalten kann in SystemC-AMS in der genutzten Fassung nur mit Differentialgleichungen, enthalten in den Grundelementen Wider-

stand, Spule und Kondensator, beschrieben werden. Mit dem konservativen Knotenmodell `elec_wire` steht eine elektrische Komponente für bidirektionale Verknüpfungen zur Verfügung, von der sich durch Vererbung andere Domänen ableiten lassen.

Für den aus Feder, Masse und Dämpfer bestehenden Vibrationssensor ist damit eine Transformation der mechanischen (v, F, k, m, c) in elektrische Größen (U, I, L, C, R) erforderlich und möglich. Die Eingangsgröße des Sensors ist ein Weg. In der Analogie entspricht jedoch die Stromstärke der Geschwindigkeit und damit der Ableitung des Weges. Daraus folgt die Notwendigkeit, das Eingangssignal vor der Zuführung zu den RLC-Gliedern zu differenzieren.

In VHDL-AMS sind die Federkonstanten mittels der Tuningspannung veränderbar. SystemC-AMS bietet in der vorliegenden Version beim Analogon Spule keine Änderungsmöglichkeit der Induktivität zur Laufzeit, so dass bei dieser Modellierung die Federkonstanten als unveränderlich angesehen werden und damit die Tuningspannung unberücksichtigt bleibt. Für die Komponente Sensor ergibt sich somit ein Aufbau wie in Bild 4 dargestellt.

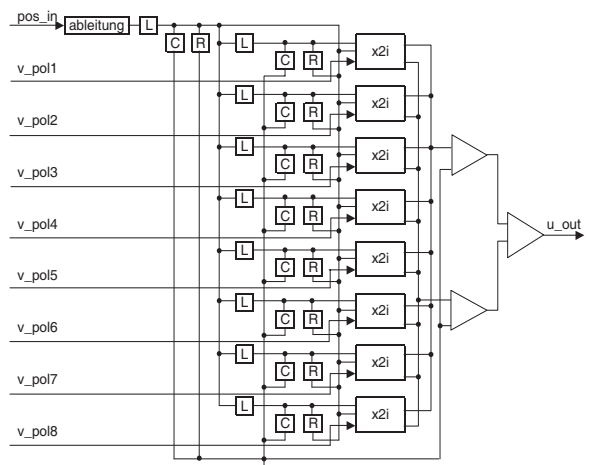


Bild 4: Struktur der Sensor-Komponente [5]

Die Auslenkungs-Strom-Transducer `x2i` wandeln in VHDL-AMS die Auslenkungen in Stromwerte um. In SystemC-AMS setzen diese Komponenten die Ausgangsspannung der Schwingkreise (als Analogon zur mechanischen Geschwindigkeit) in Ströme um. Listing 1 zeigt die Beschreibung einer Feder-Masse-Dämpfer-Komponente in SystemC-AMS.

```

ELSDF_MODULE(fmd) {
    // ----- ports -----
    elec_port in;
    elec_port out;
    elec_port gnd;
    sdf_inport<double> v_tun; //not used
    // ----- parameter -----
    sdf_para<double> m,d,k;
    // ----- components -----
    R* r1; L* l1; C* c1;
    // ----- constructor -----
    ELSDF_CTOR(fmd)
    {
        l1 = new L(this->k); //spring
        l1->a(pos_m);
        l1->b(pos_in);

        c1 = new C(this->m); //mass
        c1->a(pos_m);
        c1->b(gndm);

        r1 = new R(this->d); //damper
        r1->a(pos_m);
        r1->b(pos_in);
    };
};

```

Listing 1: Beschreibung des Sensormoduls

Modul Fuzzy-Pattern-Klassifikator

Diese rein digitale Komponente gliedert sich wie in Bild 5 dargestellt in fünf Untermodule, welche durch SystemC-Prozesse repräsentiert werden.

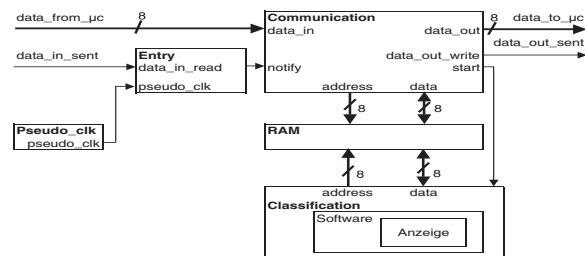


Bild 5: Struktur des Fuzzy-Pattern-Klassifikators

Pseudo_clk stellt dem Prozess Entry einen internen Takt von 1MHz zur Steuerung der Schrittgeschwindigkeit der Kommunikation mit dem Mikrocontroller zur Verfügung. Der Prozess Communication empfängt Merkmalsdaten vom Mikrocontroller und legt diese in einem internen RAM ab. Außerdem sendet er Konfigurationsdaten zurück. Die Klassifizierung ordnet die im RAM abgelegten Messwerte zwei statischen Merkmalsklassen zu. Es findet im Gegensatz zu dynamischen Merkmalsklassen kein Lernprozess während der Laufzeit statt.

Die internen Algorithmen des Moduls können in [4] und [5] nachgelesen werden. Listing 2 zeigt den Klassifikationsprozess als reinen C++-Code,

welcher in das SystemC-Modul eingebettet ist.

```

void classifier::processing(){
    for ( int k=0; k <= (anz_klass-1); k++){
        for ( int i=1; i <= 4; i++){
            index = k*(anz_merk*8+2)+(i-1)*8+2;
            xhilfe[i] = x2[i] - klass[index+1];
            s = 1.0;

            for ( int i=2; i <= 4 ; i++){
                index = k*(anz_merk*8+2)+(i-1)*8+2;
                z = tan(klass[2+index])/sqrt(s);
                wn = sqrt(1.0 + pow(z,2));
                xh = (xhilfe[1] + zw*xhilfe[i])/wn;
                xhilfe[i]=(-z*xhilfe[1]+xhilfe[i])/wn;
                xhilfe[1] = xh;
                s = s + pow(tan(klass[index+2]),2);}

            s = 0.0;
            for ( int i=1; i <= 4 ; i++){
                index = k*(anz_merk*8+2)+(i-1)*8+2;
                xh = xhilfe[i];
                if (xh >= 0.0) j = 1;
                else j = 0;
                xh = fabs(xh/klass[index+5+j]);
                xh = pow(xh,klass[index+7+j]);
                xh = xh*(1.0/klass[index+3+j] - 1.0);
                s = s + xh;}

            s = 1.0/(1.0 + s/4.0);
            cout<<"Sympathie Klasse "<<
                klass[k*(anz_merk*8+2)+1]<<" :
                "<<s<<endl;
        }
    }
}

```

Listing 2: Beschreibung des Fuzzy-Pattern-Klassifikators [8]

Innerhalb der Prozesse kann der volle C++-Sprachumfang genutzt werden. Die Bedeutung der Sprachkonstrukte ist in der SystemC-Sprachreferenz [3] angegeben.

5 Simulationsergebnisse

Da bei der Modellierung mit SystemC-AMS die Tuningspannung unberücksichtigt bleibt, wurde im VHDL-AMS-Modell die entsprechende Verbindung ebenfalls aufgetrennt. Die Bilder 6 und 7 zeigen die Verläufe am Sensorausgang bei Anlegen eines Sprunges bei identischen Simulationsvorgaben (Höhe 1µm, Schrittweite 10ns, Trapez-Integrationsalgorithmus).

Zur Einordnung der Ergebnisse sei nochmals darauf hingewiesen, dass die genutzte SystemC-AMS-Version für Kommunikationssysteme geschaffen wurde und mikromechanische Problemstellungen erst in einer späteren Version direkt unterstützt werden sollen.

In beiden Simulationsvarianten zeigt sich ein ähn-

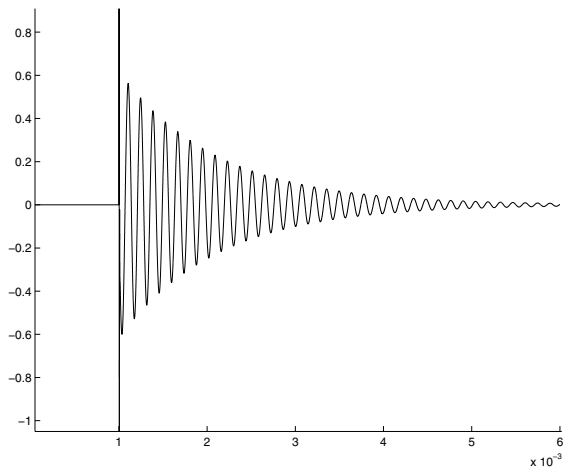


Bild 6: Sprungantwort in SystemC-AMS

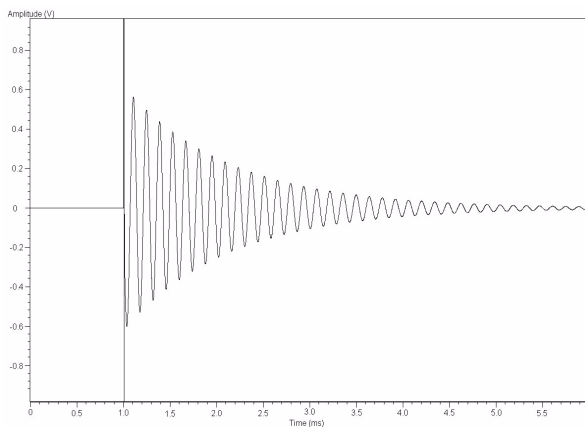


Bild 7: Sprungantwort in VHDL-AMS

liches Verhalten. Jedoch weichen die Amplituden der Schwingung um bis zu 2% voneinander ab. Ursache dieser Abweichung ist das um einen Takt verzögerte Bereitstellen der Ableitung des Eingangssignales in SystemC-AMS, während VHDL-AMS das nichtlineare Gleichungssystem löst. Die SystemC-AMS-Variante mit statischem Datenfluss beansprucht deutlich weniger Rechenzeit (2min 24s) als die VHDL-AMS-Implementierung (24min 2s) auf einer SunBlade 2900 (2 × Ultra-SPARC-III+ 900MHz, 4GB RAM, Solaris8).

Die Ergebnisse des digitalen Fuzzy-Pattern-Klassifikators differieren kaum. Tabelle 1 zeigt die ermittelten Zugehörigkeiten zu den Merkmalsklassen bei einer Folge wechselnder Beschleunigungen am Eingang.

Die Differenzen zwischen den Simulationsergebnissen beider Beschreibungssprachen sind für diese Anwendung vernachlässigbar klein.

Tabelle 1: Ergebnisse der Klassifikation

	VHDL-AMS	SystemC-AMS	Δ
Klasse 1	0,0070686	0,0070696	0,01 %
Klasse 2	0,0344848	0,0345094	0,07 %
Rechenzeit	59 min 5 s	8 min 23 s	

6 Zusammenfassung

Im vorliegenden Beitrag wurde ein Demonstrator mit einem mikromechanischen Vibrationssensor-Array in SystemC-AMS modelliert. Anwendungsgebiet der ersten Version dieser Sprache sind Kommunikationssysteme, die Nutzung zur Beschreibung von MEMS ist von der FhG IIS Dresden noch nicht implementiert.

Die Ergebnisse der SystemC-AMS-Simulation weisen im analogen Teil geringe Abweichungen in der Amplitudenhöhe gegenüber einer VHDL-AMS-Vergleichssimulation auf. Die Rechenzeit verringert sich deutlich. Bei Einbindung weiterer Löser [2], die für spätere Entwicklungsphasen geplant sind, ist mit einer Verkleinerung der Abweichungen zu rechnen.

Durch die Unterstützung des vollen SystemC- bzw. C++-Sprachumfangs bietet SystemC-AMS für digitale Komponenten vielfältige Möglichkeiten bei geringem Beschreibungsaufwand. Zusätzlich können Softwareanteile direkt mit simuliert werden. Bei analogen Komponenten hingegen fehlen noch Konstrukte besonders im nichtelektrischen Bereich, so dass der Beschreibungsaufwand im analogen Bereich gegenüber VHDL-AMS ansteigt.

7 Dank

Die hier vorgestellte Arbeit entstand im Rahmen des Teilprojektes A2 „Systementwurf“ des von der DFG geförderten SFB 379 „Mikromechanische Sensor- und Aktorarrays“.

Ein spezieller Dank gilt Herrn Karsten Einwich (FHG Dresden) für die Zusendung der SystemC-AMS-Bibliothek, Version 0.12.

8 Literatur

- [1] Einwich, K. et. al.: *White Paper SystemC-AMS Study Group*. <http://www.ti.cs.uni-frankfurt.de/systemc-ams/>

- [2] Einwich, K.: *SystemC-AMS Steps towards an Implementation*. Proceedings Forum on Specification & Design Languages FDL '03, Frankfurt/Main 2003, ISSN 1636-9874
- [3] Open SystemC Initiative (OSCI): *SystemC 2.0.1 Language Reference Manual*. 2003, <http://www.systemc.org>
- [4] Schlegel, M.; Herrmann, G.; Müller, D.: *Eine neue Hardware-Komponente zur Fuzzy-Pattern-Klassifikation*. Dresdner Arbeitstagung Schaltungs- und Systementwurf DASS 2004, Dresden, April 2004
- [5] Michel, M.: *Untersuchung der Möglichkeiten zur Beschreibung und Simulation heterogener Systeme unter SystemC-AMS im Vergleich zu VHDL-AMS*. Diplomarbeit, TU Chemnitz, Juni 2004
- [6] 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
- [7] SFB 379: *Arbeits- und Ergebnisbericht 2001-2003*. Teilprojekt A4 „Mehrbereichs-Sensorrarrays zur Bewegungsanalyse auf der Basis der oberflächennahen Si-bulk-Mikromechanik“, TU Chemnitz, 2003
- [8] Bocklisch, S. F.: *Prozeßanalyse mit unscharfen Verfahren*. Verlag Technik, Berlin 1987
- [9] Kazmierski, T. J.; Al Junaid, H.: *Synchronization of analogue and digital solvers in mixed-signal simulation on a SystemC platform*. Proceedings Forum on Specification and Design Languages FDL'03, pp. 65 - 71, Frankfurt/Main 2003