

B. Däne / V. Duridanova / W. Fengler

Konzept und Entwurfsmethodik für ein eingebettetes System zur hochleistungsfähigen Informationsverarbeitung in Nanopositionier- und Messmaschinen

1. Einleitung

Nanopositionier- und Messmaschinen [1] sind komplexe Einrichtungen, in denen mechanische, optische, elektronische und informationsverarbeitende Komponenten in optimaler Weise zusammenwirken müssen, um das gewünschte Ergebnis zu erreichen. Der hier betrachtete informationsverarbeitende Teil wird in Form eines hochleistungsfähigen eingebetteten Systems als Mehrprozessorverbund aus Digitalen Signalprozessoren realisiert. Die Genauigkeits- und Geschwindigkeitsparameter des Gesamtsystems werden durch die Leistungskenngrößen der Informationsverarbeitung entscheidend mitbestimmt. Für die Hard- und Software des eingebetteten Systems wird eine Entwurfsmethodik benötigt, welche das Verhalten des gesamten Systems in den Entwurfsprozess einbezieht.

2. Defizite des bisherigen Forschungsstandes

Gegenwärtige leistungsfähige Entwurfsverfahren beruhen auf der teilweisen oder vollständigen Modellierung des zu entwerfenden eingebetteten Systems und seiner einbettenden Umgebung. Dabei tritt regelmäßig das Problem auf, dass sehr unterschiedliche Modellierungsdomänen miteinander verknüpft werden müssen. Insbesondere ist eine gleichberechtigte Integration kontinuierlicher und diskreter Betrachtungsweisen notwendig. Bei der Mehrzahl der existierenden Verfahren ist entweder keine wirklich domänenübergreifende Modellierung möglich, oder es ist kein durchgehender Entwurfsweg bis hin zur Implementierung verfügbar.

3. Methodischer Lösungsansatz

Die beiden Ziele, einerseits eine Entwurfsmethodik und andererseits einen Prototypen für das eingebettete System zu entwickeln, werden in enger Wechselwirkung und gegenseitig befruchtender Weise angegangen. Auf der einen Seite entsteht ein Toolverbund, welcher einen durchgängigen Entwurf der benötigten Hard- und Software ermöglicht. Dieser unterstützt den kompletten Entwurfsweg, von der Spezifikation über Modellierung und formale Analyse bis hin zur Implementierung des Hard- und Softwaresystems. Die Modellierungs- und Analyseebene ermöglicht die vollständige Einbeziehung des Verhaltens der mechanischen, optischen und leistungselektronischen Komponenten, da nur auf diese Weise eine zuverlässige Beurteilung der Leistungs- und Qualitätskenngrößen des Gesamtsystems möglich ist.

Als Modellierungsmittel kommen hochentwickelte grafische Darstellungsformen zum Einsatz, wie objektorientierte hybride und zeitbehaftete Petri-Netze [2], gefärbte Zustands- und Se-

quenzdiagramme [3][4] und hierarchische Multi-Domain-Umgebungen. Ein Entwurfsframework wird diesen Verbund unterschiedlicher Entwurfsmittel zu einem durchgängig nutzbaren Gesamtsystem mit Eignung für das Co-Design von Hardware, Software und nichtelektronischen Komponenten integrieren.

Der Prototyp für das einsatzbereite eingebettete System ist als Mehrprozessorsystem mit Digitalen Signalprozessoren der Familie TMS320C6000 [5] ausgelegt, was die Implementierung komplexer Strategien zur effizienten Ressourcenzuteilung notwendig macht. Daraus folgt unter anderem die Weiterentwicklung des problemangepassten Echtzeitbetriebssystems eRTOS [6][7], welches im Gegensatz zu typischen kommerziellen Lösungen zum Beispiel die Priorisierung des Durchsatzes ratenmonoton geschedulter Tasks gegenüber der Reaktionszeit auf externe Ereignisse ermöglicht.

4. Aktuelle Ergebnisse zur Modellierungsmethodik

4.1. Modell für ein Atomkraftmikroskop

In einer aktuellen Arbeit [8] wurde beispielhaft ein heterogenes simulierbares Modell für ein Atomkraftmikroskop geschaffen. Das Modell umfasst die Dynamik des mechanischen Systems mit den elektromagnetischen Antrieben, die Wechselwirkung von Antastsystem und Prüfling, die Funktion der inkrementellen laseroptischen Wegsensoren (einschließlich simulierbarer Störeinflüsse), die Messwertaufbereitung und –auswertung (mit Interpolation und Fehlerkompensation), die Algorithmen zur dreidimensionalen Lageregelung und die Ablaufsteuerung des Bewegungsvorganges.

Zur Modellierung wurde die hierarchische Multi-Domain-Entwurfsumgebung „MLDesigner“ eingesetzt [9][10]. Es wurden Modellkomponenten aus unterschiedlichen Modellierungsdomänen kombiniert. Insbesondere wurden quasikontinuierliche (synchroner Datenfluss) und ereignisdiskrete Modellierungsparadigmen kombiniert. Zu ersteren zählt der größte Teil der Komponenten in den Regelkreisen, zu letzteren z.B. die Richtungserkennung und Periodenzählung innerhalb der Sensorsignalaufbereitung sowie die Ablaufsteuerung für den Bewegungsvorgang.

Das Modell vereint in Software zu realisierende Funktionen mit solchen, welche dem mechanischen und optischen System sowie der elektronischen Hardware zuzuordnen sind. Es ist mit Registrierungs- und Anzeigekomponenten für Auswertungszwecke instrumentiert. Im Ergebnis von Simulationsläufen wurden Daten gewonnen, welche unter anderem das Verhalten der Regelkreise und die Ergebnisse kompletter Messabläufe darstellen.

4.2. Detaillierte Beschreibung des Modells

Bild 1 zeigt die oberste Ebene des gesamten Modells. Bei den hier sichtbaren Blöcken handelt es sich überwiegend um Module, die einerseits dem mechanischen und optischen System und andererseits der Software des eingebetteten Systems zuzuordnen sind. Es sind jedoch auch Eingabe- und Anzeigeelemente für den Simulationsvorgang vorhanden, wie zum Beispiel die Blöcke „XMgraph.“, „TkShowValues.“, „Stop Button“ und einige andere.

Bemerkung: Die Bilder finden Sie am Ende dieser Datei. Sämtliche Bilder beruhen auf Ergebnissen der Arbeit [8] und wurden mit „MLDesigner“ [9] erzeugt.

Das Modul „nano_scanner“ (Bild 2) enthält das Modell des mechanischen Systems. Es wird mit

den Stellgrößen für die drei Achsen angesteuert und liefert die Ist-Werte für die Position aller drei Achsen sowie einen Wert für die Position der Abtastnadel. Dazu ist ein dynamisches Modell für den beweglichen Teil („3d_system“), ein Modell für den Antastvorgang („cantilever“) sowie ein Modell für das Oberflächenrelief des Prüflings („oberflaeche“) enthalten. Die beiden ersten beruhen auf zugänglichen Informationen über derartige Systeme [11][12], letzteres erzeugt im simulierten Beispiel eine einfache sinusartige „Hügelstruktur“.

Das Modul „laser_noise“ (Bild 3) modelliert das optische System mit der laserinterferometrischen, inkrementell arbeitenden Wegmessung. Während die Module „interferometer_*“ die idealen Sinus- und Cosinus-Signale der drei Achsen erzeugen, werden in den sich anschließenden Blöcken „*_noise“ unterschiedliche Fehlereinflüsse simuliert. Die Stärke der Fehlereinflüsse kann während der Simulation entweder manuell oder per Script variiert werden.

Das Modul „filter_dig_pos_*“ (Bild 4), das für jede der drei Achsen existiert, enthält die in Software zu realisierende Auswertung der Sensorsignale. Hier werden die Sinus- und Cosinus-Signale nach Verstärkungskorrektur und Tiefpassfilterung einem Block „dig_interpolator“ (Bild 5) zugeführt, welcher pro Periode vier auswertbare Impulsflanken liefert. Der sich anschließende Block „up_down_count“ (Bild 6) realisiert die Erkennung der Bewegungsrichtung und die Impulzzählung. Der Zählerstand verkörpert einen Teil des Messwertes.

Im Bild 4 wird weiterhin aus den Sinus- und Cosinus-Werten im Block „position_arctan“ ein Feinwert für die Position gewonnen, welcher die Auflösung der Wegmessung erhöht. Zur Sicherung der Konsistenz zwischen Grob- und Feinwert sind dann noch einige Korrekturen nötig.

Das in Bild 1 für jede Achse vorhandene Modul „*_pos_regler“ (Bild 7) enthält den Regleralgorithmus für die Lageregelung der drei Achsen. Es handelt sich um einen parametrierbaren PID-Regler, dessen Struktur für dieses Modell aus vorhandener Software abgeleitet wurde und dessen Parameter im Modell optimiert wurden.

Die Sollwerte für die x- und y-Achsen kommen von der Ablaufsteuerung in „trigger_alternate“ und „triangle_gen“. Im vorliegenden Beispiel wird eine rasterförmige Abtastung der Oberfläche des Prüflings realisiert. Für die z-Achse wird zunächst die Stellgröße erhöht, bis der erste Kontakt der Messspitze mit der Prüflingsoberfläche auftritt. Anschließend wird die z-Achse so geregelt, dass die Position der Messspitze (bzw. die hier wirkende Kraft) konstant bleibt. Die Ist-Werte an der z-Achse beschreiben nun das gesuchte Oberflächenprofil.

4.3. Ergebnisse der Simulation

Während der Simulation können die Werteverläufe der einzelnen Variablen aufgezeichnet und anschließend ausgewertet werden. Bild 8 zeigt z.B. die Istwerte der Positionen über der Modellzeit. Da eine rasterförmige Abtastung vorgenommen wurde, zeigt die x-Achse (rot) eine ständige Hin- und Herbewegung, während sich die y-Achse (blau) schrittweise weiterbewegt. Man erkennt eine gewisse Unsicherheit der Positionierung sowie gelegentliches Überschwingen.

Die z-Achse (grün) zeigt zu Beginn eine Zustellbewegung. Anschließend folgt die Kurve dem Relief der Prüflingsoberfläche. Die Nadelposition (schwarz) soll durch den Regler der z-Achse konstant gehalten werden. Man sieht hier Abweichungen und Einschwingvorgänge, die jeweils durch Bewegungen der x- und y-Achse angestoßen werden. Bei Abweichungen der Nadelposition tritt ein entsprechender Fehler bei der z-Achse auf.

Bild 9 zeigt einen kleinen Ausschnitt des vorigen Bildes. Man erkennt deutlich die stufenweise

Veränderung des x-Wertes (rot) und des y-Wertes (blau) während des Abtastvorganges. Überlagert ist eine Schwingung entsprechend der Eigenfrequenz des mechanischen Systems.

Bild 10 zeigt den Ausschnitt eines anderen Simulationslaufes, während dem die simulierten Fehler im optischen System variiert wurden. Im mittleren Bereich (Modellzeit 0.32s bis 0.48s) war die Amplitude der zufälligen Störungen („Rauschen“) erhöht. Man erkennt deutliche Abweichungen und Unsicherheiten insbesondere bei der Position in x-Richtung.

Das resultierende Oberflächenprofil für den Simulationslauf aus Bild 8 ist in Bild 11 zu sehen. Hier wurden die registrierten z-Werte über der x-y-Ebene aufgetragen, so dass die „Hügelstruktur“ der Oberfläche sichtbar wird. Bild 12 zeigt dieselbe Darstellung für einen anderen Simulationsablauf, für den die Auflösung des Abtastrasters in x- und y-Richtung jeweils um den Faktor vier erhöht wurde. Erkennbar sind die Einflüsse der angesprochenen Störungen auf das Ergebnis.

Mögliche Erweiterungen des Modells wären z.B. andere Varianten der Ablaufsteuerung, insbesondere eine Steuerung der Bewegungsschritte in x-y-Richtung in Abhängigkeit vom Erreichen des Sollbereiches bei der Nadelposition. Weiterhin könnte durch entsprechende Nachbearbeitung der Einfluss der Schwingungen der z-Achse auf das Ergebnis verringert werden.

4.4. Ansätze zur Implementierung

In einer weiteren Arbeit [13] entstand ein Konzept für die Implementierung von softwaretechnisch zu realisierenden Funktionen, welche als Teil eines Modells in MLDesigner vorliegen, in ausführbare Software für das eingebettete System. Dabei wurde wiederum dem Umstand Rechnung getragen, dass quasikontinuierliche und ereignisdiskrete Modellierungsdomänen zu kombinieren sind.

Die heterogenen Schedulingalgorithmen im Betriebssystem *e*RTOS [6] lassen sich dafür gut verwenden. Für in quasikontinuierlichen Domänen modellierte Systembestandteile werden ratenmonotone Tasks („kooperierende Tasks“) eingesetzt. Die Schedulingtabelle für diese Tasks wird entsprechend der gegenseitigen Verhältnisse zwischen den im Modell vorhandenen Abtastraten generiert.

Für ereignisdiskret modellierte Bestandteile werden preemptive Tasks („nichtkooperierende Tasks“) verwendet. Diese laufen nur auf Anforderung und können verdrängt werden. Die Weitergabe der Ereignisse erfolgt über das Nachrichtensystem [7] von *e*RTOS. Für die Kommunikation zwischen den beiden Realisierungsdomänen wird ebenfalls das Nachrichtensystem benutzt. Die Anbindung von Ein- und Ausgabebaugruppen erfolgt über Treiberkomponenten, welche eine Schnittstelle für hardwarenah realisierte Ein- und Ausgaberoutinen darstellen.

5. Ziele und Ausblick

Die nächsten Arbeitsschritte betreffen unter anderem die Entwicklung der Mehrprozessorfähigkeit für das Betriebssystem *e*RTOS und dessen Portierung auf die veränderten Hardware-Plattformen [14]. Parallel dazu sind die begonnenen Arbeiten zur Modellierungsmethodik zu verallgemeinern und zu erweitern. Dabei wird angestrebt, die verschiedenen Ansätze in einem gemeinsamen methodischen Rahmen zusammenzuführen.

Das hier beschriebene Projekt wird in Kooperation mit zahlreichen Partnern unterschiedlicher Fachrichtungen betrieben. Das gemeinsame Ziel ist die Realisierung hochgenauer Nanopositio-

nier- und Messmaschinen sowie die Schaffung von Methoden und Kompetenzen, die eine effektive Weiterentwicklung und Verwertung der Ergebnisse ermöglichen.

Literatur- und Quellenhinweise:

- [1] Gerd Jäger, Eberhard Manske, Tino Hausotte: Nano Positioning and Measuring Machine. In: euspen 2001, European Society for Precision Engineering and Nanotechnology, Second International Conference, Turin 2001
- [2] Vesselka Duridanova, Thorsten Hummel: Modelling of Embedded Mechatronic Systems Using Hybrid Petri Nets. In: IASTED International Conference Modelling, Identification, and Control, Feb. 19-22, 2001, Innsbruck, Austria. IASTED/ACTA Press 2001, vol. 1, pp. 521-526
- [3] Olga Fengler, Wolfgang Fengler, Vesselka Duridanova: Modeling of Complex Automation Systems Using Colored State Charts. In: ICRA 2002: Proceedings of the 2002 IEEE International Conference on Robotics and Automation. IEEE 2002. vol. 2, pp. 1901-1906
- [4] Olga Fengler, Thorsten Hummel, Wolfgang Fengler: Modellierung kooperierender Prozesse mit gefärbten Sequenzdiagrammen. In: 5. GI/ITG/GMM-Workshop: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen. Tübingen, 25.-27.02.2002. Shaker Aachen 2002, S. 199-208.
- [5] Naim Dahnoun: Digital Signal Processing Implementation Using the TMS320C6000 DSP Platform. Prentice Hall 2000
- [6] Christian Uhle, Bernd Däne, Todor Vangelov, Wolfgang Fengler: Neue DSP-Hardware- und Softwarelösungen für den Einsatz in Mehrkoordinaten-Nanomeß- und Positioniersystemen. 45. Internationales Wissenschaftliches Kolloquium, TU Ilmenau 2000
- [7] Sebastian Schmidt: Konzeption und Realisierung eines dynamisch konfigurierbaren Echtzeitbetriebssystems für den Einsatz z in einem Messwerterfassungssystem. Diplomarbeit Nr. 2001-11-01/0011/IN95/2231, TU Ilmenau 2001
- [8] Kirke Rimbach: Modellierung eines Messsystems mit dem Modellierungswerkzeug ML-Designer. Studienarbeit, TU Ilmenau (IA/RA) 2002
- [9] Homepage der MLDesign Technologies, Inc., Palo Alto: <http://mldesigner.com/>
- [10] Horst Salzwedel: MLDesigner. Der nächste Schritt in Electronic Design. In: Workshop Werkzeuge und Ansätze für Spezifikation und Synthese komplexer Systeme. Ilmenau, 30. November 2001
- [11] S. Salapaka, M. Dahleh: Friction in Atomic Force Microscopy: Modeling & Control. Technical Report Series CCEC-99-0913, University of California, Santa Barbara 1999.
- [12] N. A. Burnhamy et al.: How does a tip tap? In: Nanotechnology 8 (1997) 67-75.
- [13] Stefan Schramm: Konzeption und Realisierung einer Implementierungskomponente aus dem Modellierungstool ML-Designer in ein DSP-System mit dem Betriebssystem eRTOS2. Diplomarbeit Nr. 2002-09-02/037/IN95/2231, TU Ilmenau 2002
- [14] Marc Leverenz: Konzeption und Realisierung von Betriebssystemsoftware für ein Dual-DSP-Board. Diplomarbeit Nr. 2002-06-03/024/IN95/2231, TU Ilmenau 2002

Autorenangaben:

Dr. Ing. Bernd Däne

Dr. Ing. Vesselka Duridanova

Prof. Dr. Ing. habil. Wolfgang Fengler

Technische Universität Ilmenau, PSF 100565

98684 Ilmenau, Germany

Tel.: +49-3677-69-2825

Fax: +49-3677-69-1614

E-mail: bernd.daene@tu-ilmenau.de

Web: <http://tin.tu-ilmenau.de/ra/>

Teile der hier dargestellten Arbeit wurden gefördert durch das Thüringer Ministerium für Wissenschaft, Forschung und Kunst unter FKZ B509-00002. Die Weiterführung wird im DFG-Sonderforschungsbereich Nr. 1923 erfolgen.

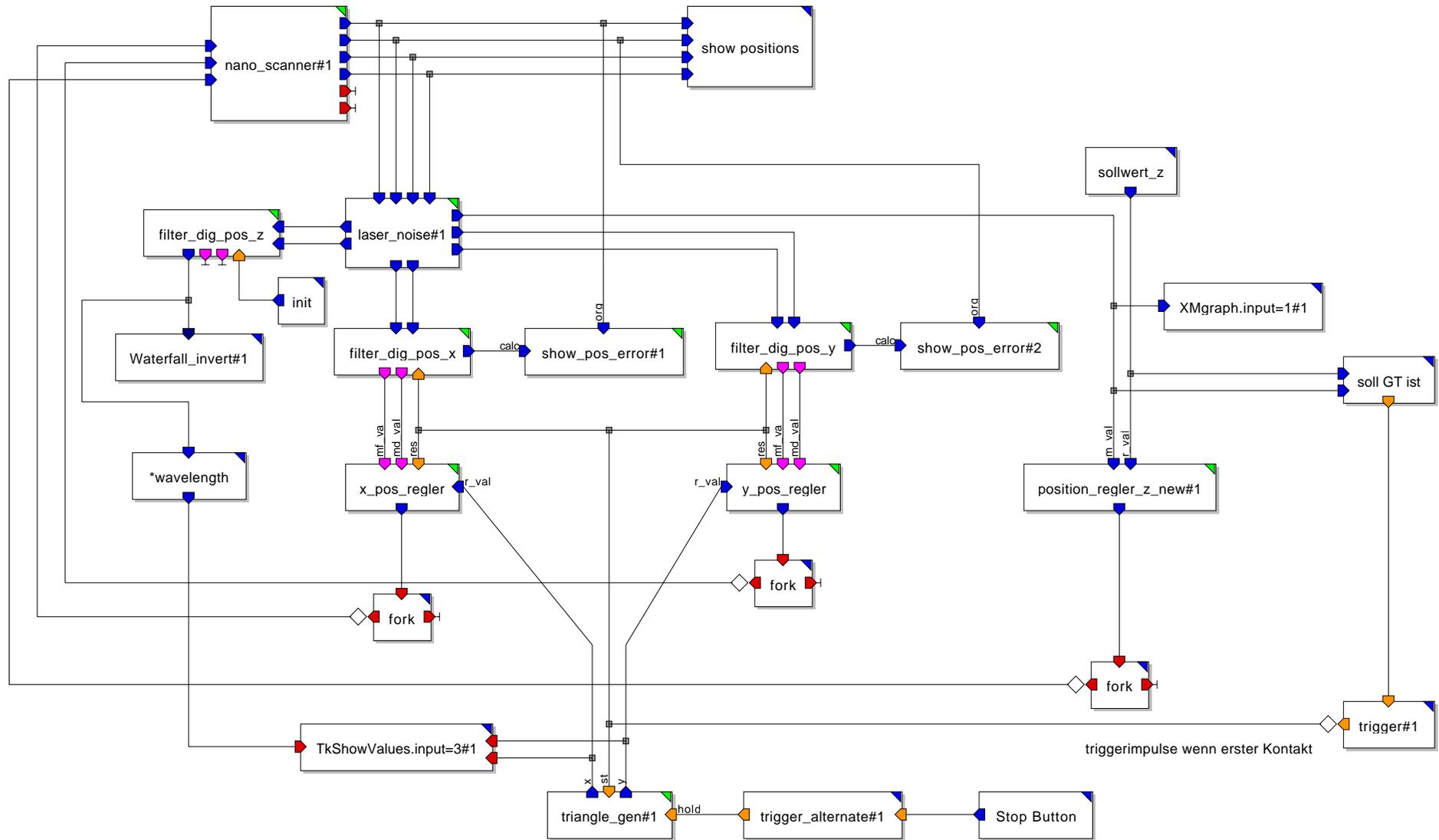


Bild 1: Gesamtmodell

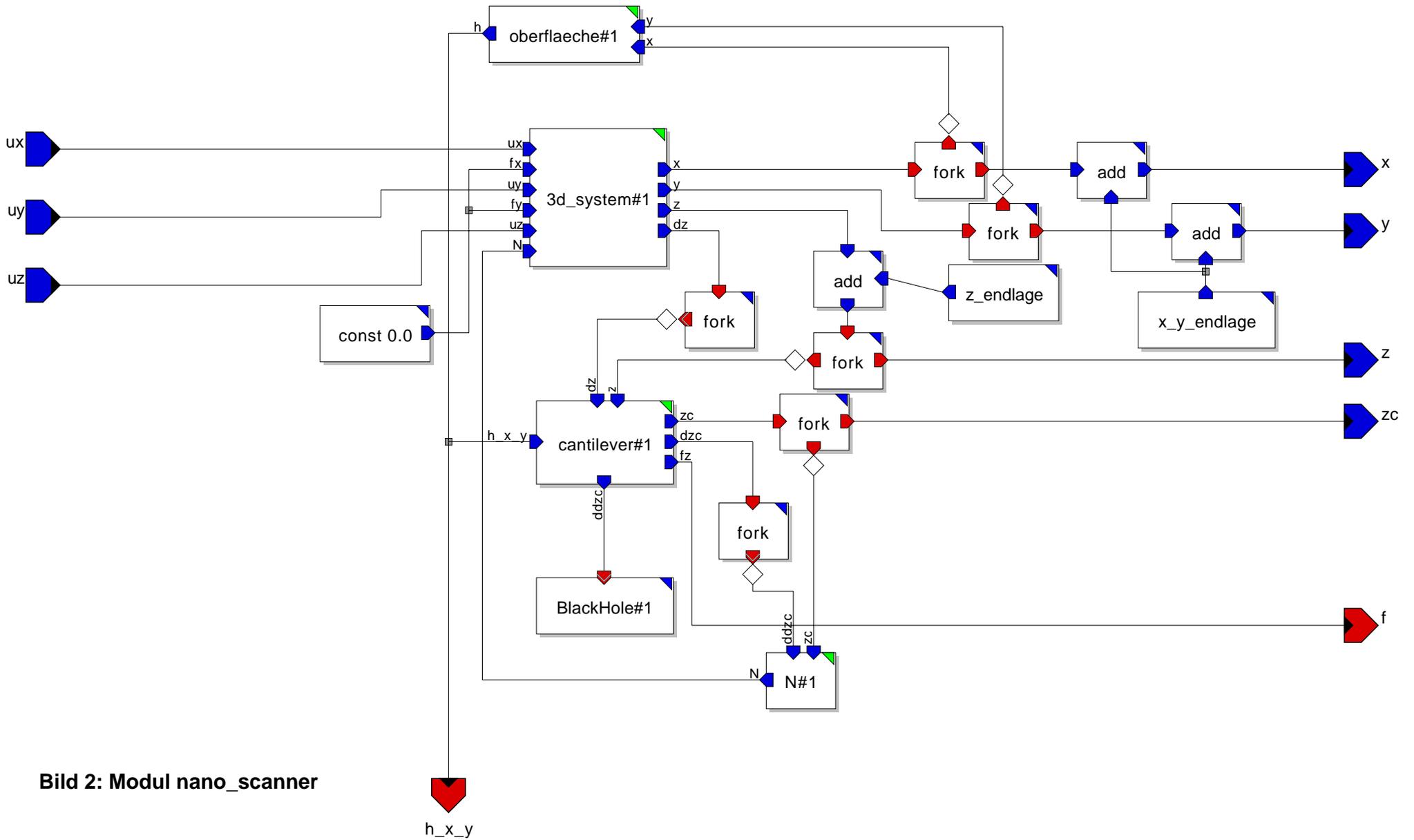


Bild 2: Modul nano_scanner

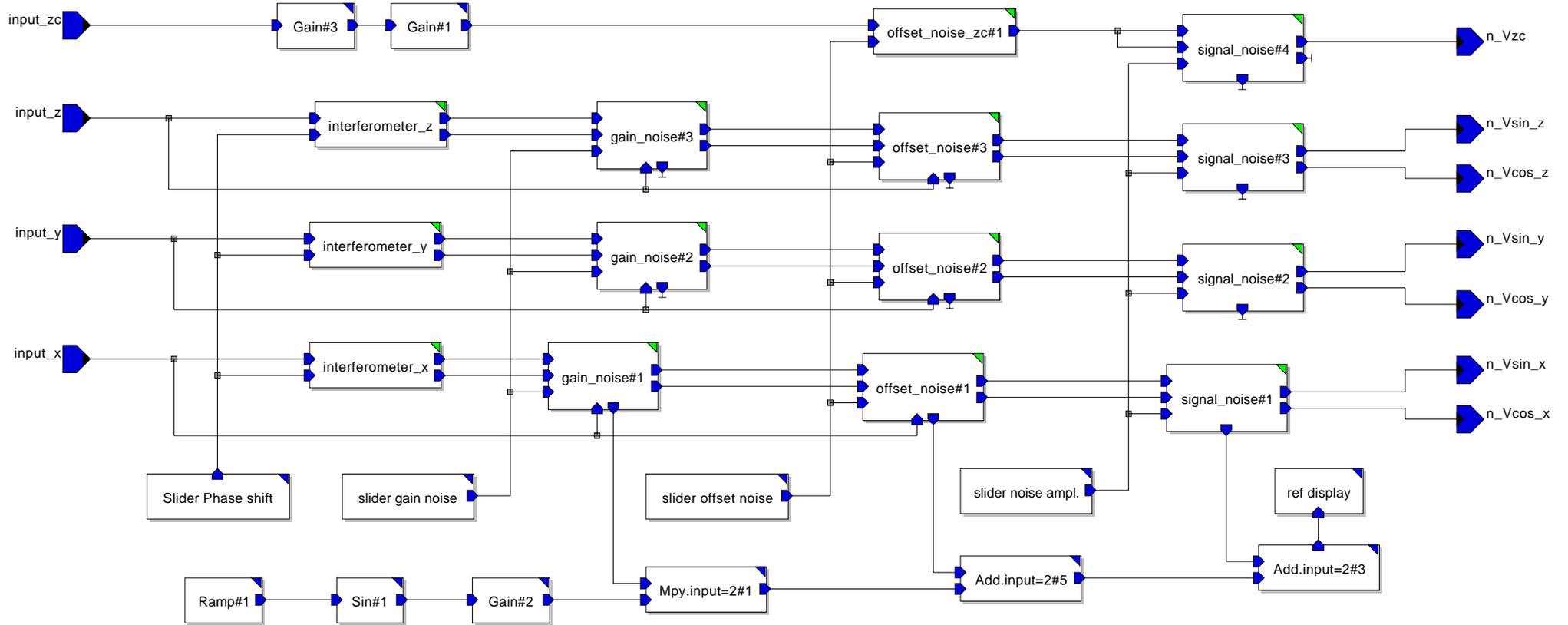


Bild 3: Modul laser_noise

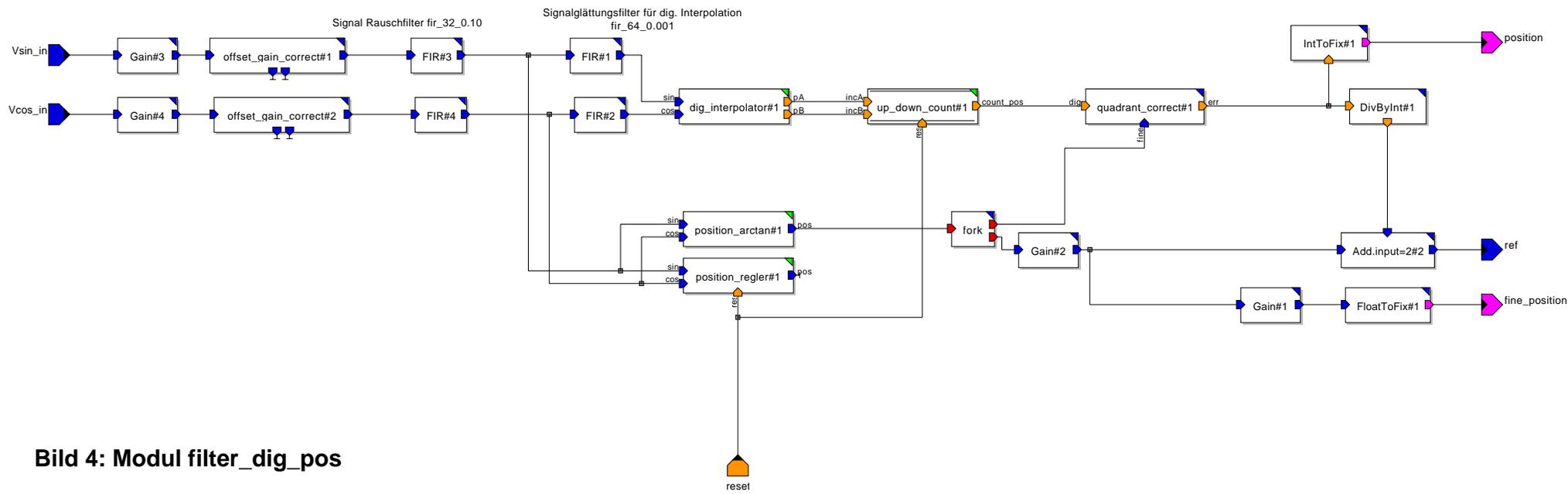


Bild 4: Modul filter_dig_pos

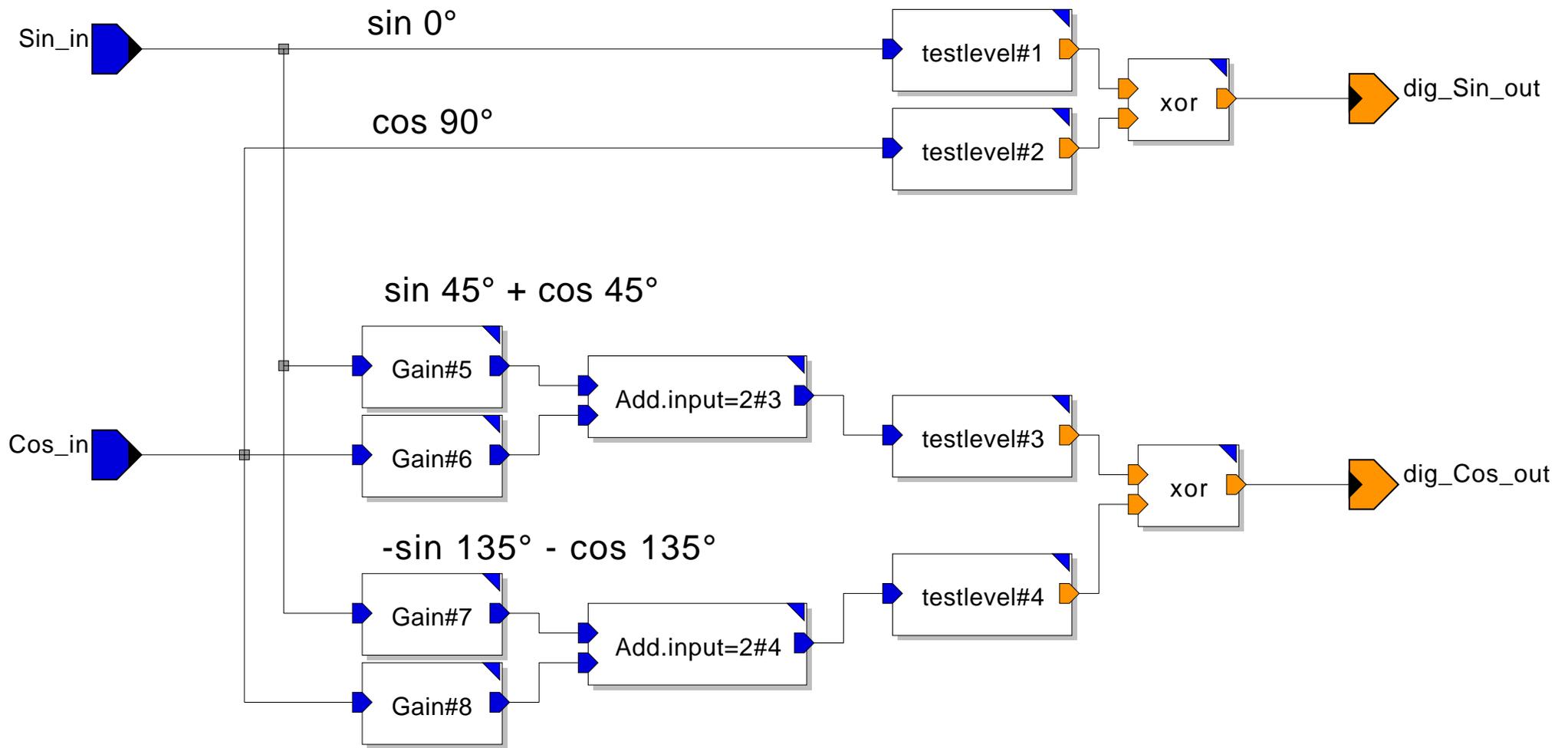


Bild 5: Modul dig_interpolator

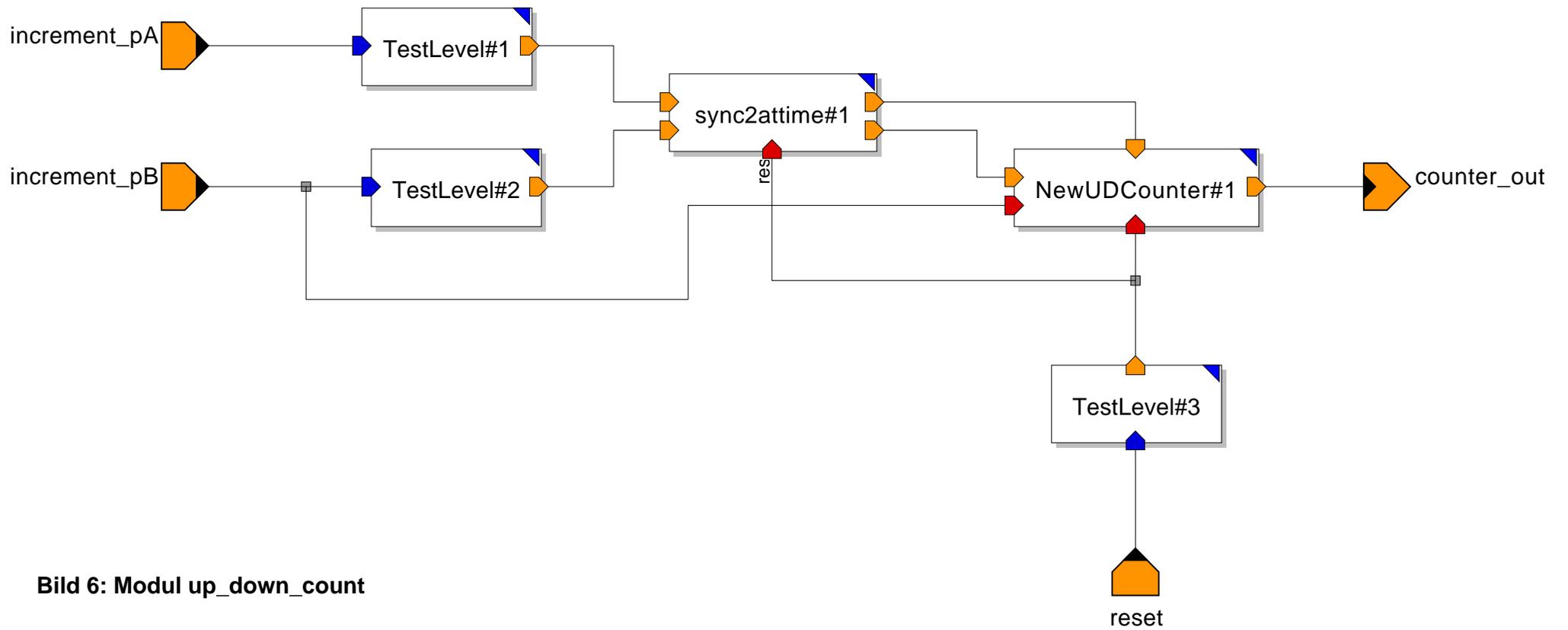


Bild 6: Modul up_down_count

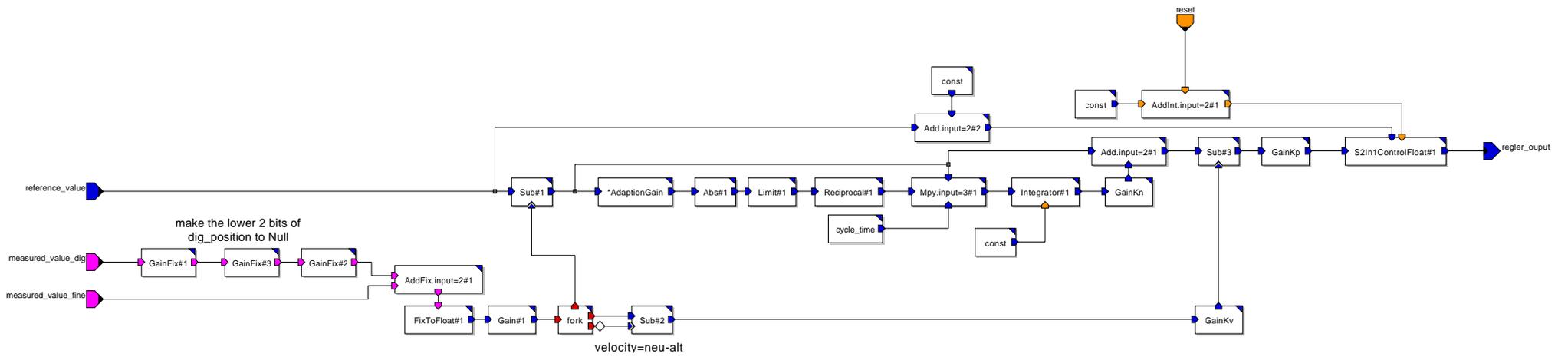


Bild 7: Modul pos_regler

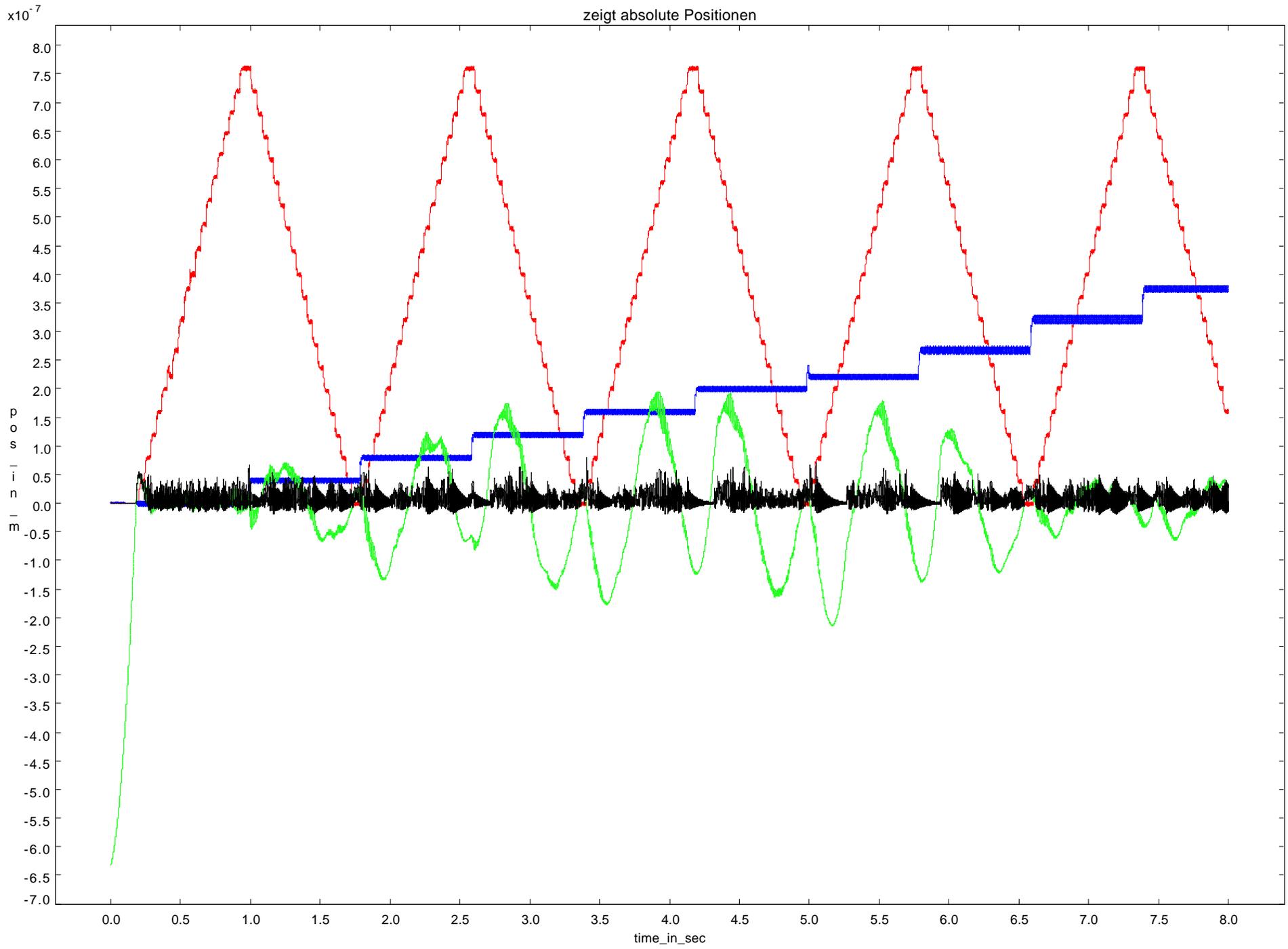


Bild 8: Zeitverlauf der Positionswerte

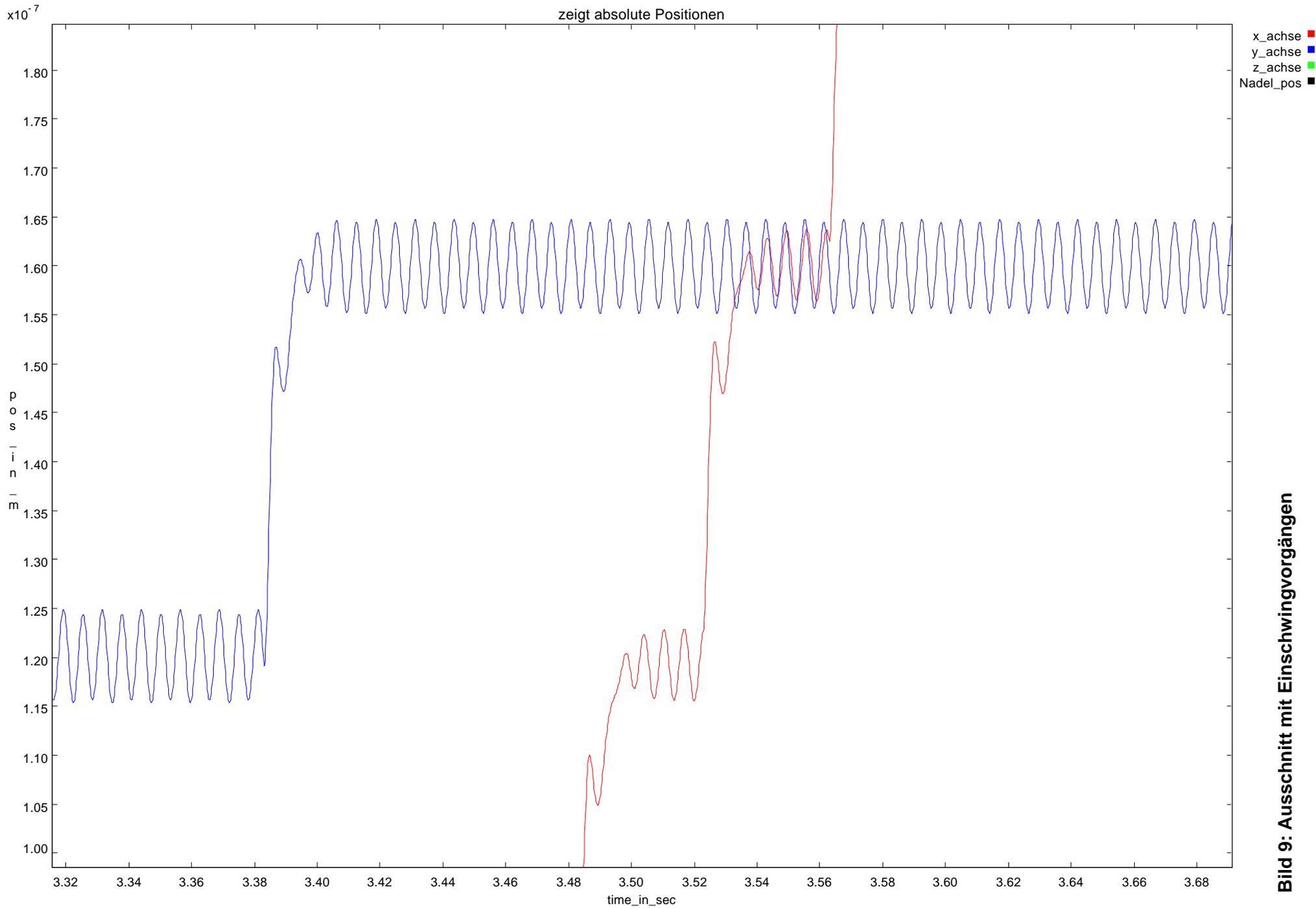


Bild 9: Ausschnitt mit Einschwingvorgängen

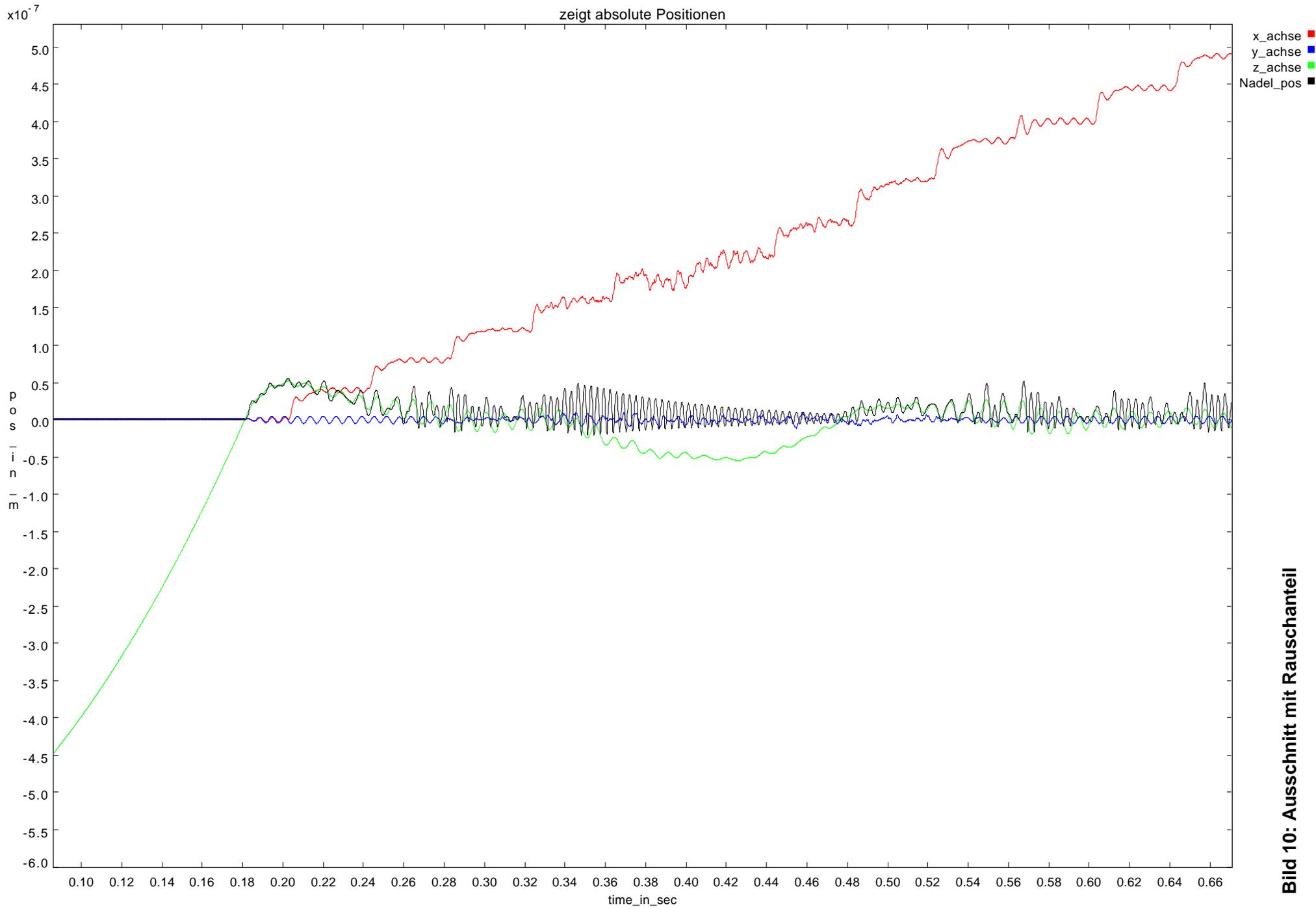


Bild 10: Ausschnitt mit Rauschanteil

modell.Waterfall_invert#1

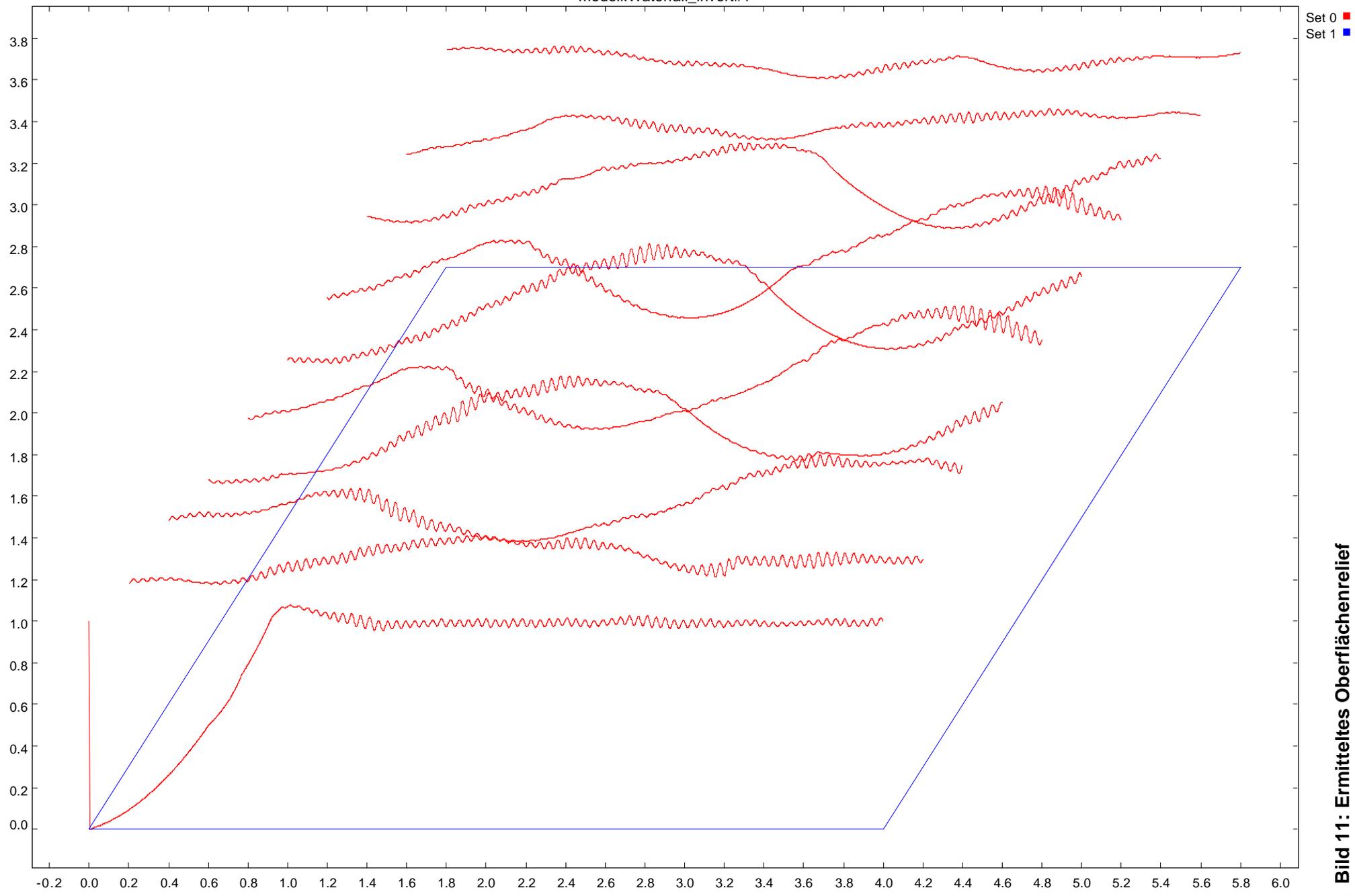
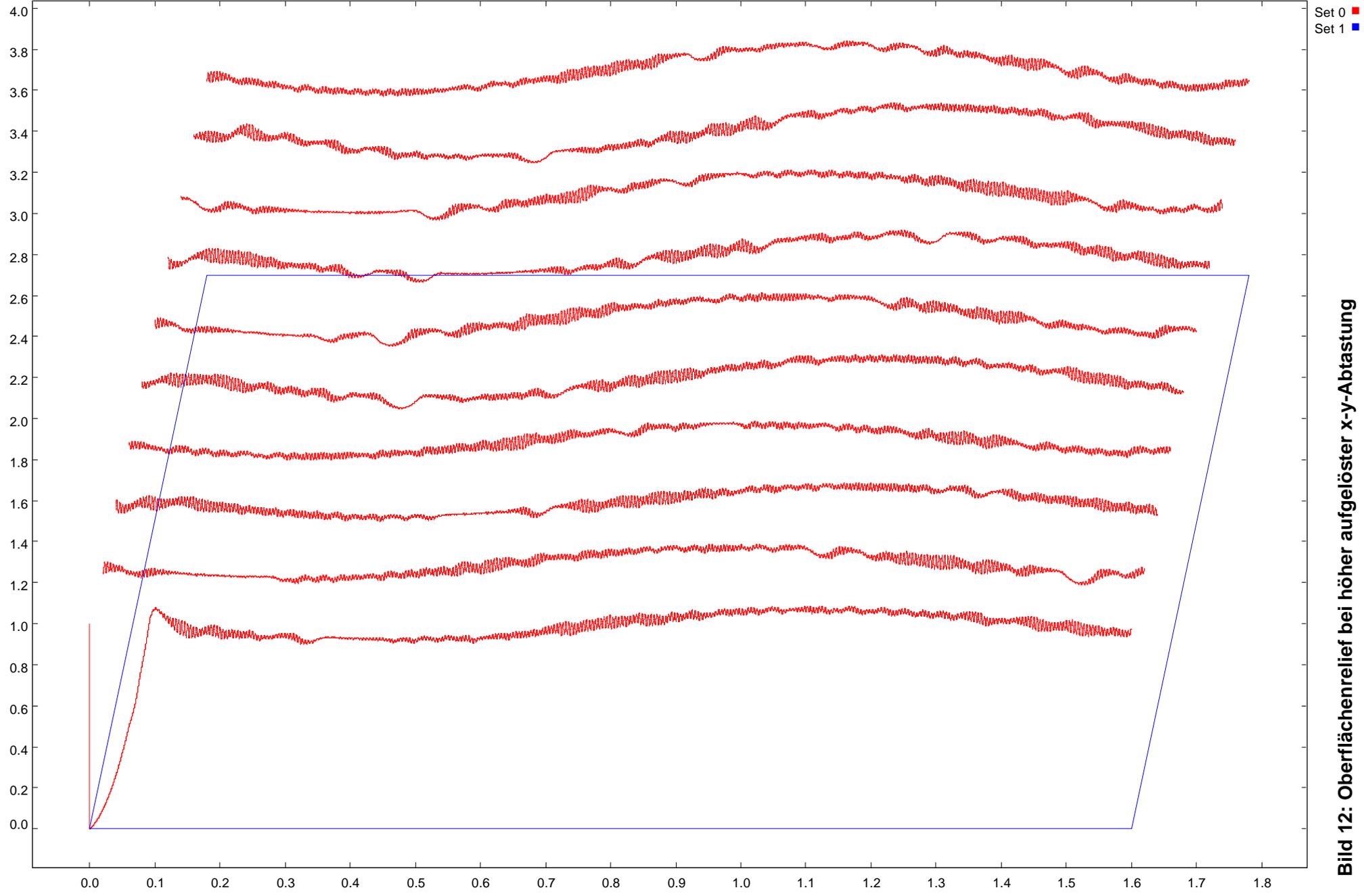


Bild 11: Ermittertes Oberflächenrelief

$\times 10^{-2}$

modell.Waterfall_invert#1



Set 0 ■
Set 1 ■

Bild 12: Oberflächenrelief bei höher aufgelöster x-y-Abtastung

$\times 10^3$