

ANSÄTZE ZUR REALISIERUNG DER BLOCKORIENTIERTEN PARALLELEN SIMULATIONSMODELLE FÜR DYNAMISCHE NETZOBJEKTE

Lapko V.V., Moldovanov A.V., Svjatnyj V.A.
Fakultät für Rechentchnik und Informatik (FRTI)
Nationale Technische Universität Donezk, Artemstr. 58, 83000, Ukraine
{lapko | mold | svjatnyj}@cs.dgtu.donetsk.ua

Abstract

Lapko V.V., Moldovanov A.A., Svjatnyj V.A. Approaches to implementation of block-oriented parallel simulation models for dynamic network objects. Development of parallel simulation models of complex network objects and other dynamic systems requires special software components. The article considers some approaches to block-oriented implementation of virtual parallel simulation models and their devirtualisation on the example of a network object.

Einführung

Die Qualitätssicherung von Automatisierungsprojekten für komplexe Netzobjekte wird durch die eine Rechnersimulation der entwickelten Lösungen auf allen Etappen von industrienahen Projekte gewährleistet. Die modellgestützte Projektierung [1] benötigt Werkzeuge für die rechnergestützte Projektierung, eine effiziente Modellierung und die Simulation komplexer dynamischer Systeme. Der Bereich „Dynamische Netzobjekte“ beinhaltet topologisch komplexe technische Netze, deren Automatisierungssysteme nach dem Dekompositionsprinzip entwickelt werden und bis zu drei bis vier Hierarchie-Ebenen haben. Für die Modellierung auf der niedrigsten Prozess- und Reglerebene wird die blockorientierte Simulationssprache MATLAB-Simulink benutzt. Auf höheren Ebenen entsteht wegen des Kompositionsverfahrens eine Komplexitätsschwelle der Modellierung von einem dynamischen Verhalten, die für die Simulationslösungen höhere Rechenleistungen sowie die rechnergestützte topologieabhängige Erstellung der Simulationsmodelle benötigt. Diese Lösungen werden im Rahmen der parallelen Simulationsumgebung (PSU) [2] realisiert. Die Erfahrung zeigt, dass für die Erstellung der parallelen Simulationsmodelle von komplexen Netzobjekten sowie anderen dynamischen Systemen PSU spezielle Softwarekomponenten gebraucht werden. In diesem Beitrag werden verschiedene Ansätze zur

blockorientierten Herleitung von virtuellen parallelen Simulationsmodellen und deren Devirtualisierung am Beispiel von Netzobjekten betrachtet.

Blockorientierte Modellierung der dynamischen Netzobjekte und Automatisierungssysteme

Die Subsysteme und Module der projektierten Automatisierungssysteme, die durch eine Dekomposition im CASE-Werkzeug erstellt wurden, werden in Simulink modelliert und untersucht. Dabei stehen dem Entwickler folgende Simulink-Vorteile zur Verfügung: Simulink erlaubt die Entwicklung eigener Blöcke in C und MATLAB; Modelle können mit äußeren Programmen zusammenarbeiten; einfache und bequeme Archivierung, Modularisierung, Erstellung von Modellbibliotheken; benutzerfreundliche Bedienoberfläche. Um diese Vorteile auch bei der parallelen MIMD-Modellierung von komplexen Netzobjekten zu nutzen, wird die folgende rechnerunterstützte Vorverarbeitungsetappe vorgeschlagen:

1. Topologiebeschreibung der Netzobjekte und deren Automatisierungssysteme; die Netztopologie wird als gerichteter Graph $G(U, V)$ mit den Mengen von Knoten $|U| = n$ und Kanten $|V| = m$ dargestellt;
2. Anwendung des Topologieanalysators [3] für eine formale Darstellung der topologischen Eigenschaften des simulierten Objektes (Vektoren der Luftströme $Q(X, Y)$, $X = (X1, X2, \dots, Xi)$, $Y = (Y1, Y2, \dots, Yi)$ des Graphenbaums und Antibaums, Inzidenz- und Schleifenmatrizen A, S , die Parametermatrizen K, R , Vektor H der Ventilatorcharakteristiken, Vektor Z der Nichtlinearitäten $Qi^*|Qi|$);
3. Automatische Generierung der Objektgleichungen in der für Simulink notwendigen Form. Ursprüngliches Gleichungssystem des Netzobjektes

$$A \times Q = 0$$

$$K \times \frac{dQ}{dt} = S \times H - S \times R \times Z \quad (1)$$

wird vom Gleichungsgenerator bezüglich der gesuchten Vektoren bzw. ihrer Ableitungen gelöst:

$$X = -W \times Y$$

$$\frac{dY}{dt} = S_U \times H - S_U \times R \times Z \quad (2)$$

Hier sind $W = A_X^{-1} \times A_Y$ und $S_U = (S_Y \times K_Y - S_X \times K_X \times W)^{-1} \times S$.

4. Rechnerunterstützte Erstellung der Simulink-Modellstruktur mit den Vektor-Matrix-Blöcken;
5. MIMD-Realisierung des blockorientierten Simulationsmodells.

Ansätze zur Realisierung der parallelen Simulationsmodelle

Es werden die folgende Ansätze zur Realisierung der parallelen Modelle vorgeschlagen:

1. „Block - Prozess - Prozessor“: jedem Simulink-Block entspricht ein Prozess, der im Prozessor des MIMD-Systems erfüllt wird. Die Modelloptimierung und eine Lastbalancierung der Prozessoren werden durch eine Definition von gleich komplexen Blöcken und der Suche der gleichbelasteten Block-Ketten realisiert. Diese Block-Ketten werden im Rahmen der einzelnen Prozesse berechnet und den einzelnen Prozessoren zugeordnet. Dieser Ansatz wurde in [4] untersucht.

2. SPMD-Ansatz: es wird eine Vektor-Matrix-Darstellung des blockorientierten Simulationsmodells entwickelt und das erhaltene Blockdiagramm entsprechend der gesuchten Variablen repliziert. Die Replikation wird in zwei Varianten vorgeschlagen.

3. Replikation-1. Parallelisierungsansatz „Replikation-1“ wird auf folgende Weise durchgeführt. Der Gleichungsgenerator, außer Hauptoperationen der Formierung von Matrizen A , S , K , R , W und S_U , führt noch eine Replikationsoperation durch, die komponentenweise das Gleichungssystem (2) formiert. Entsprechend den Zeilen des Gleichungssystems wird ein Simulink-Blockdiagramm für jede gesuchte Variable entwickelt. Die Parallelisierung wird durch die Implementierung jedes Blockschaltbildes in dem ihm zugeordneten Prozessor des MIMD-Systems erreicht. Dieses Prinzip ist wegen der Fähigkeit der Simulink-Modelle, mit den äußeren Programmen zu funktionieren, realisierbar: die in den MIMD-Prozessoren implementierten Simulink-Exemplare kommunizieren im Rahmen des parallelen Lösungsalgorithmus und berechnen die gesuchten Komponenten der Vektoren. Der Datenaustausch zwischen den Blöcken von Simulink-Exemplaren wird mit Hilfe der eigenen Simulink-Mittel durchgeführt. Die Datenübertragung zwischen den Prozessoren kann mit Hilfe von Message Passing Interface (MPI) realisiert werden. Für diese Operation sind die Schnittstellen notwendig, die eine Verbindung zwischen der Simulink-Umgebung und MPI verwirklichen. Die Verwendung von MPI-Standard erlaubt die plattformunabhängige Modellrealisierung bei dem Datenaustausch zwischen den Prozessoren. Betrachten wir diese Variante der Replikation an einem Beispiel des Netzobjektes (Bild 1).

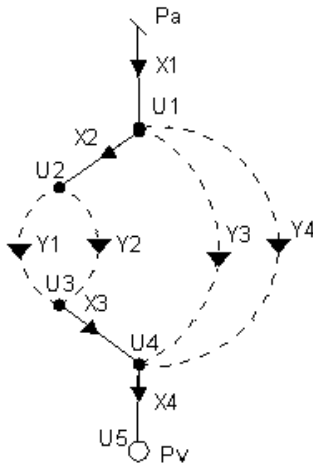


Bild 1: Graph des Netzobjektes

Simulink-Blockdiagramm des den Gleichungen (2) relevanten Simulationsmodells wird in Bild 2 dargestellt. Die Matrix W und ihre Zeilen W_{x1}, \dots, W_{x4} spielen die Rolle des Kommutators, der die geforderten Elemente des Vektors Y an Eingänge des jeweiligen Summierers anschließt. Die Matrix $S_U = (S_Y K_Y - S_X K_X W)^{-1} S$ für das konkrete Netzobjekt muß vor der Entwicklung des Modellblockdiagramms erhalten werden. Elemente dieser Matrix spielen zwei Rollen: a) als arithmetische Elemente, die die während der Lösung von (2) bezüglich \bar{Y} topologisch umgewandelten Matrizen S, K, W beinhalten; b) als Kommunikationselemente, die entsprechende Komponenten von Vektoren $\bar{H}, \bar{RZ}, \bar{RRZ}$ zu den Eingängen der Summierer SUM5 – SUM8 einschließen (RR ist diagonale Matrix der regelbaren Widerstände, dabei die Komponenten $RR_{x3}, RR_{y1}, \dots, RR_{y4}$ liefern entsprechende Reglermodelle). Simulink-Modell (Bild 2) beinhaltet $n-1$ X-Ketten $W_{xi} \rightarrow SUM_i \rightarrow ABS_i \rightarrow MUL_i \rightarrow R_{xi}$ (algebraischer Teil) und γ Y-Ketten $S_{uyj} \rightarrow INT_j \rightarrow ABS_j \rightarrow MUL_j \rightarrow R_{yj}$ (integro-algebraischer Teil).

Die X- und Y- Ketten werden den Prozessoren $\Pi_{y1}, \dots, \Pi_{y\gamma}$ zugeordnet. In den Netzobjekten mit $\gamma = n - 1$ wird die volle paarweise Berechnung der X- und Y-Vektorkomponenten erreicht. In dem Fall $\gamma > n - 1$ haben die Prozessoren $\Pi_{x1}, \dots, \Pi_{xn-1}$ die gleiche Last mit den Prozessoren $\Pi_{y1}, \dots, \Pi_{yn-1}$, dabei realisieren übrige $\gamma - (n - 1)$ Prozessoren nur gebliebene Y-Ketten. Falls $\gamma < n - 1$, werden die Prozessoren $\Pi_\gamma - \Pi_{(n-1)-\gamma}$ zusätzlich die Komponenten $X_{\gamma+1} \dots X_{n-1}$ berechnen.

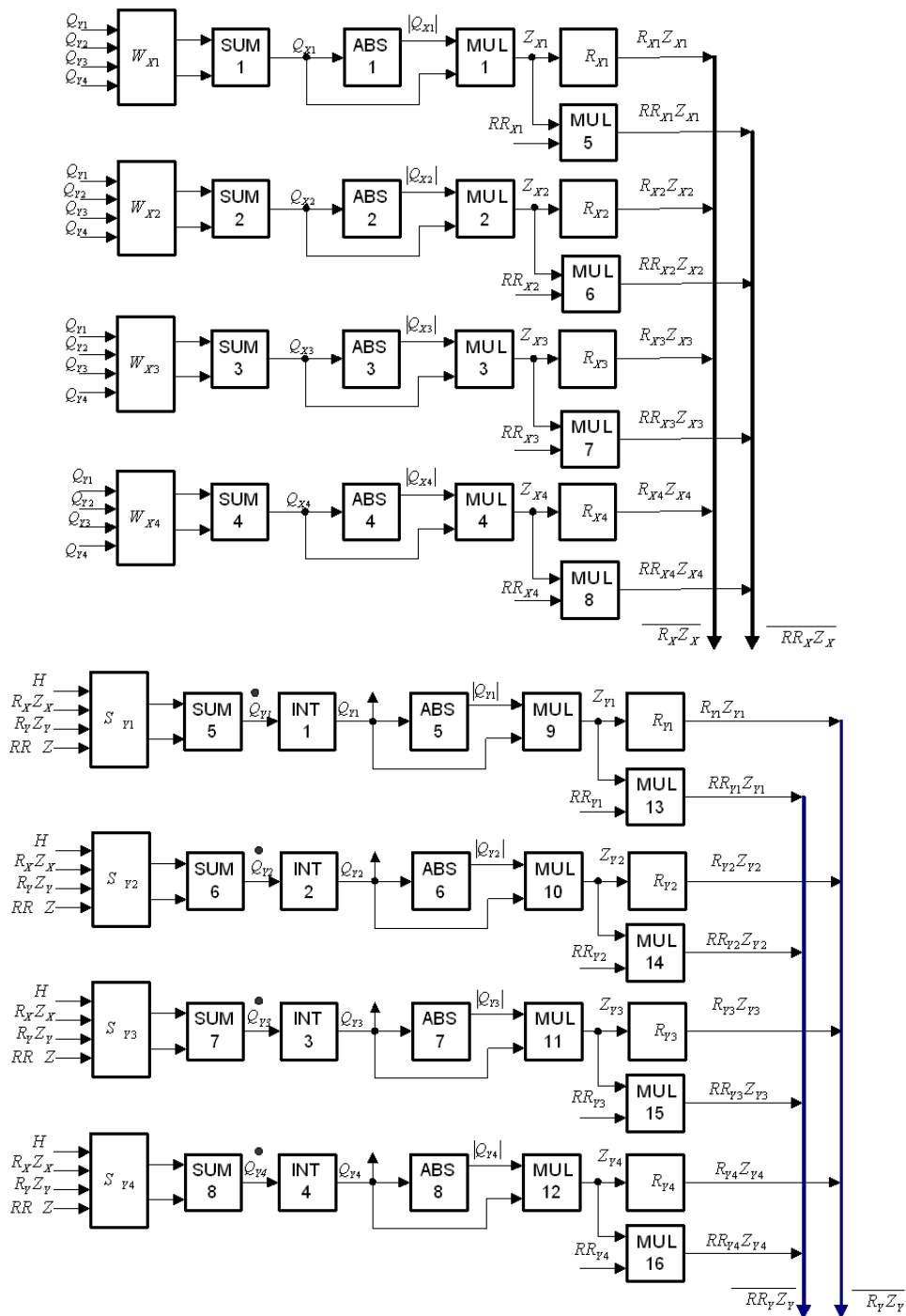


Bild 2: Block-Diagramm des Simulink-Modells

4. Replikation – 2. Diese Variante der SPMD-Replikation sieht einen Prozessor für jede Block-Kette vor. Dieser Ansatz wird auf folgende Weise realisiert:

- 1) Die Operation $\bar{X} = -W\bar{Y}$ wird in $n-1$ Blöcke nach einer Komponentenanzahl des Vektors \bar{X} repliziert.

- 2) Die Operation $\bar{Y} = \int (S_{\gamma} \bar{H} - S_{\gamma} R \bar{Z} - S_{\gamma} R R \bar{Z}) dt$ wird in γ Integrierer repliziert.
- 3) Der Block \bar{Z} enthält einen X -Teil ($n-1$ Multiplikationsoperationen von Typ $X_i | X_i, i = \overline{1, n-1}$) und einen Y -Teil (γ Multiplikationsoperationen von Typ $Y_j | Y_j, j = \overline{1, \gamma}$). Auf diese Weise werden die Ketten „Block X – Block Z_x “, „Block Y – Block Z_y “ verfolgt.
- 4) Die rechte Seite der zweiten Gleichung im System (2) wird so dargestellt:

$$(S_y K_y - S_x K_x W)^{-1} [(S_x H_x + S_y H_y) - S_x R_x Z_x - S_y R_y Z_y - S_x R R_x(t) Z_x - S_y R R_y(t) Z_y]$$
Dabei werden die Operationen bezüglich der Vektoren X und Y getrennt. Im Blockdiagramm werden X -Schleifen und Y -Schleifen herausgezogen. Nach dem Prinzip der SPMD-Funktionierung wird die X -Schleife i vom Prozessor i sowie die Y -Schleife j vom Prozessor j vollständig erfüllt ($i = \overline{1, n-1}, j = \overline{1, \gamma}$).

Zusammenfassung

Die vorgeschlagenen Ansätze zur Realisierung von blockorientierten parallelen Simulationsmodellen für dynamische Netzobjekte werden in der an der TU Donezk entwickelten parallelen Simulationsumgebung benutzt und liefern einen wichtigen Beitrag zur Erhöhung der Benutzerfreundlichkeit bei der rechnergestützten Projektierung und Simulation komplexer Automatisierungssysteme.

Literatur

1. Svjatnyj, V.A., Lapko, V.V., Moldovanov, A.V.: Integration von rechnergestützten Projektierung und Simulation von Automatisierungssystemen. In: A. Kuhn und S. Wenzel (Hrsg.): Simulationstechnik. 11. Symposium in Dortmund, November 1997, Vieweg, 1997, S. 510- 514.
2. Svjatnyj, V.A., Anoprienko, A.J., Bräunl, T., Reuter, A., Zeitz, M.: Massiv parallele Simulationsumgebung für dynamische Systeme mit konzentrierten und verteilten Parametern. In: G. Kampe und M. Zeitz (Hrsg.): Simulationstechnik. 9. Symposium in Stuttgart, Oktober 1994, Vieweg, 1994, S. 183-188.
3. Pererva, A.A.: Topologischer Analysator der parallelen Simulationsmodelle von dynamischen Netzobjekten. In: Informatik, Kybernetik und Rechentechnik. Collected Volume of Scientific Papers. Donetsk State Technical University. Donetsk, 1999, S.73-78 (in Russisch).

4. Moldovanov, A.V., Sviatnyi, S.N.: Blockorientierter Ansatz zur Herleitung paralleler Modelle für dynamische Systeme. – Simulationstechnik, 13. Symposium in Weimar, September 1999. Tagungsband, S. 319-324.
-

Дата надходження до редколегії: 4.12.2003 р.