

Inhaltsverzeichnis

1. Einleitung	6
1.1 Ziel und Aufbau der Arbeit	6
1.2 Einführung in die Mechatronik	9
1.3 Das Konzept des Mechatronik Laboratorium Paderborn	10
1.4 Computer-Aided Mechatronics Laboratory - Ein Labor im Rechner	11
2. Vorkenntnisse und Stand der Technik der Prozeßkopplung in der Mechatronik	12
2.1 Grundbegriffe der Prozeßdatenverarbeitung	12
2.2 Kopplung von technisch-physikalischen Prozessen mit den informationsverarbeitenden Systemen	13
2.2.1 Die Echtzeitbedingungen	13
2.2.2 "Intelligente" Sensoren	14
2.2.3 Das CoSyMa-Konzept	15
3. Ein Konzept zur verteilten Simulation mit Hardware-in-the-Loop	17
3.1 Die Konzept-Struktur	18
3.2 Die experimentorientierte Darstellung	18
3.2.1 Modellierte Basisexperimente - MBE	20
3.2.2 Technische Basisexperimente - TBE	21
3.2.3 Die Experimentschnittstelle - ES	22
3.3 Die rechnerorientierte Darstellung	23
3.3.1 Modellierte Recheneinheit - MRE	26
3.3.2 Technische Recheneinheit - TRE	27
3.3.3 Netzwerkkontrolleinheit - NKE	28
3.4 Die Echtzeitbedingungen in der rechnerorientierten Darstellung	29
3.5 Eine Modellierung der rechnerorientierten Darstellung als Petri-Netz	31
3.5.1 Die Theorie der Petri-Netze: Begriffserklärungen	32
3.5.2 Stellen/Transitions-Netze: S/T-Netze	33
3.5.3 Die Modellierung der Recheneinheiten	36
3.6 Die Kommunikation zwischen Recheneinheiten als DIN/ISO- Basis-Referenzmodell	42
4. Hardwareaspekte der Implementierung	44
4.1 Vorstellung des Transient-Werkzeugs	44
4.1.1 Softwareaspekte von Transient	44
4.1.2 Initialisierung des Transputernetzwerkes	45

4.1.3	Erzeugung und Auswahl der Basissysteme	46
4.1.4	Auswahl der Integratoren	46
4.1.5	Verteilung der Rechen- und Kommunikationslast	46
4.1.6	Codegenerierung	47
4.1.7	Einhalten der Echtzeitbedingungen	47
4.2	Das Transputer-Konzept von INMOS	48
4.2.1	Das "Dual Inline Transputer Module"- TRAM	51
4.2.2	Die "Module Motherboard Architecture"- MMA	52
4.3	Die Abstimmung der INMOS-Philosophie für das Konzept zur verteilten Simulation mit Hardware-in-the-Loop	52
4.3.1	Das virtuelle "Dual Inline Transputer Module" (vTRAM)	53
4.3.2	Die erweiterte "Module Motherboard Architecture" (eMMA)	54
4.4	Die Hardware-Abbildung der technischen Recheneinheiten	55
4.4.1	Die "Technical Experiment Description" - TED	55
4.4.2	VMEbus-Plattform	57
4.4.3	PHS-Bus-Plattform	63
5.	Anwendungsbeispiele der Prozeßkopplung mittels einer VMEbus-basierten Hardware-Plattform	64
5.1	Prüfstand zur Simulation einer aktiven Radaufhängung mit "Hardware-in-the-Loop" /Castiglioni et al. 92/	64
5.1.1	Übersicht über den KFZ-Prüfstands	65
5.1.2	Prüfstandsmodell	67
5.1.3	Funktionsprinzip des Prüfstands	73
5.1.4	Prozeßkopplung des KFZ-Prüfstands	74
5.1.5	Erprobung im Versuch und vergleichende Simulation	80
5.2	Eine Entwicklungsumgebung für ein fahrerloses Transportfahrzeug	81
5.2.1	Aufbau des fahrerlosen Transportfahrzeugs <i>Ameise</i>	81
5.2.2	Modellbildung eines fahrerlosen Transportfahrzeugs	82
5.2.3	Reglerrealisierung	84
5.2.4	Erprobung des fahrerlosen Transportfahrzeugs im Versuch	85
6.	Zusammenfassung und Ausblick	87
7.	Glossar	89
8.	Anhang	91
8.1	Die Implementierung der erweiterten "Module Motherboard Architecture" (eMMA)	91
8.2	VMEbus-Constants für die Prozeßkopplung	91

8.3 DSL-Beschreibung für die Simulation des KFZ-Prüfstands mit Hardware-in-the-Loop _____	93
8.4 DSL-Beschreibung des fahrerlosen Transportfahrzeugs _____	109
9. Literatur _____	118