

2.3 Das Simulationsprogramm TRNSYS

2.3.1 Prinzip

Durch die fortgeschrittene Entwicklung der Computertechnik ist es heute zunehmend möglich und sinnvoll, zur Erfassung und Berechnung von Energiesystemen Simulationsprogramme einzusetzen. Simuliert werden dabei vorrangig Systeme, die von mehreren voneinander unabhängigen Faktoren beeinflußt werden und bei denen nichtzyklische Speichervorgänge eine Rolle spielen.

Ziel der Simulation ist es, Vorhersagen über das Verhalten dieser Systeme zu ermöglichen. Dazu ist es zunächst notwendig, die Systeme exakt zu beschreiben. Ist dies einmal geschehen, können leicht Variantenrechnungen durchgeführt werden. Dadurch wird es möglich, neue Systeme optimal zu planen und bestehende Systeme zu optimieren.

Ein sehr vielseitiges, aber auch sehr schwer handhabbares Simulationsprogramm ist TRNSYS. Es wurde 1973 an der University of Wisconsin in Madison für die Simulation solarthermischer Anlagen entwickelt. Seitdem wurde es ständig weiterentwickelt. Gegenwärtig ist die Version 14.2 aktuell, wobei die Erstellung dieser Arbeit mit der 1993 aktuellen Version 13.1 begonnen und aufgrund des zu hohen Umstellungsaufwandes auch weitergeführt wurde.

Heute ist der Einsatz von TRNSYS besonders für die Berechnung von Solar- und Klimaanlage sowie von Systemen der Kraft-Wärme-Kopplung sinnvoll. Am Einsatz für Regelungsuntersuchungen wird gegenwärtig gearbeitet.

Dem Programm liegt die Idee zugrunde, ein großes mathematisches Problem in viele Teilprobleme zu zerlegen und sie durch Module zu beschreiben, welche dann vom Anwender in einer Steuerdatei, dem sogenannten Deck, miteinander zu verknüpfen sind. Solche Module existieren derzeit für Komponenten thermischer Energiesysteme und für die Gebäudesimulation.

TRNSYS berechnet das Systemverhalten in Zeitschritten von einer Stunde oder weniger. Für einen einzelnen Zeitschritt wird angenommen, daß konstante Bedingungen herrschen, also konstante Einstrahlungen, konstante Temperaturen und Wärmeströme usw. Dieser stationäre Zustand wird jeweils ausgerechnet. Welche Größen am Ende ausgegeben werden sollen, legt der Nutzer im Deck selbst fest.

Sind Ganzjahressimulationen notwendig, sollte die Zeitschrittweite, die sich aus den relevanten Zeitkonstanten ergibt, eine Stunde nicht um mehrere Größenordnungen unterschreiten. Die Rechnung dauert dann zu lange.

Das Programm ist in FORTRAN 77 geschrieben. Da es mit Quellcode ausgeliefert wird, kann ein Programmierer neue Module hinzufügen bzw. Vorhandene modifizieren.

TRNSYS ist kein vordergründig kommerzielles Programm, was sich auch im Preis niederschlägt. Es ist eine Größenordnung billiger als vergleichbare Programme (zum Beispiel TAS). Im Gegenzug muß eine gewisse Fehlerhaftigkeit in Kauf genommen werden, um deren Abstellung sich der Nutzer selber bemühen muß.

In TRNSYS ist im Prinzip jede mögliche energetische Schaltung berechenbar. Mit beherrschbarem Aufwand ist dies jedoch nur bei Vorliegen von Unterprogrammen für jede Komponente möglich. Der Bestand ist momentan bereits relativ groß. Einzelne Komponenten können selbständig programmiert werden.

2.3.2 Das TRNSYS-Deck

Im Deck werden die Komponenten einer energetischen Schaltung miteinander verknüpft, die Datenein- und ausgabe organisiert und für den Simulationsablauf notwendige Anweisungen vereinbart. Es verfügt über eine eigene Formatsprache. Die Komponenten nennt man in TRNSYS "Type". Es gibt Bauelemente-Types, wie Kollektor, Pumpe, Speicher usw., sowie externe Types wie Strahlungsprozessor und Datenleser.

Jede Komponente im System wird über drei Arten von Informationsflüssen beschrieben (siehe Bild 24). Das Verhalten einer Systemkomponente - beschrieben über die Outputs - hängt von Einflußgrößen, die sich im Jahresverlauf nicht ändern, definiert über "Parameter" und Einflußgrößen, die sich von Zeitschritt zu Zeitschritt ändern können, beschrieben über "Inputs" ab. Die Inputwerte werden üblicherweise von den Outputs anderer Komponenten bezogen, dürfen aber auch konstant sein.

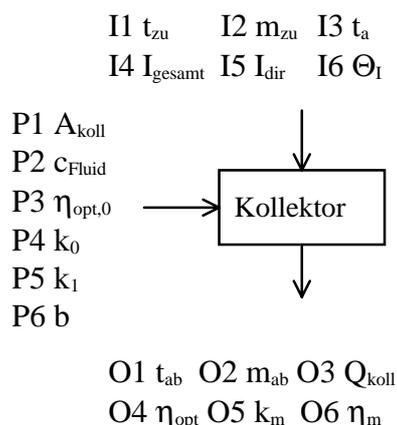
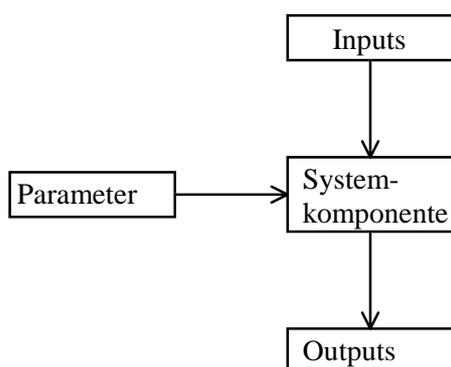


Bild 24: Allgemeine Komponentenbeschreibung

Bild 25: Belegung Sonnenkollektor

Parameter-, Input- und Outputbelegung sind für jede Komponente charakteristisch und können Katalogen /41/ entnommen werden. Erstellt der Nutzer eigene Types, kann er auch in den Grenzen der physikalischen Sinnhaftigkeit des Modells diese Belegung selbst festlegen. In Bild 25 ist beispielhaft die Belegung eines Sonnenkollektors angegeben. Parameter sind zum Beispiel die Kollektorfläche A_{koll} und der optische Wirkungsgrad bei senkrechter Einstrahlung $\eta_{opt,0}$, Inputs z. B. die Zulauftemperatur t_{zu} und die Direktstrahlung I_{dir} und Outputs z. B. Ablauftemperatur t_{ab} und die Kollektorleistung Q_{koll} .

Einige externe Types weichen von der Standardstruktur ab. So besitzt der Datenleser keine Inputs, da er die Wetterdaten direkt aus einer externen Datei bezieht und der Drucker keine Outputs, da er die Ergebnisse direkt in eine externe Datei ablegt. Bei der Ergebnistabelle tritt eine zusätzliche Kategorie auf, "Labels", die für die Beschriftung der Spaltenköpfe benötigt wird.

Um was für eine Art von Komponente es sich jeweils handelt, wird durch eine Type-Nummer festgelegt. So hat der Kollektor die Nummer 1 und der allgemeine Speicher die Nummer 4. Da in einem Deck einzelne Komponentenarten mehrfach auftreten können, zum Beispiel Speicher, besitzt jede Komponente zusätzlich noch eine Unit-Nummer, die im ganzen Deck nur einmal auftreten darf.

Neben den Types gibt es in einem Deck noch "Statements", welche Anweisungen definieren, die nicht einzelnen Komponenten zugeordnet werden können, aber für die Funktionsweise des Decks notwendig sind.

Dies betrifft zum Beispiel die Festlegung der Simulationsschrittweite, die Zuweisung von Wetter- und Ausgabedatei und die zulässige Iterationszahl je Zeitschritt.

Eine Sonderstellung nehmen die "Equations" ein, in denen Berechnungen vorgenommen werden können, die nicht einzelnen Komponenten zuzuordnen sind, zum Beispiel die Berechnung stetiger Regelsignale. Eine Equation übernimmt einen oder mehrere Outputs, berechnet daraus eine Variable, die dann von einer anderen Komponente als Input übernommen wird. Die Übernahme von Wetterdaten und gebäudecharakterisierenden Größen sowie die Übergabe von Ergebnissen erfolgt mittels Dateien.

Im folgenden soll kurz dargestellt werden, wie aus einer vorgegebenen Schaltung ein TRNSYS-Deck erarbeitet wird. Das Beispiel ist sehr einfach und seine technische Realisierung nicht empfehlenswert. Es geht nur um die Darstellung der prinzipiellen Vorgehensweise bei der Erstellung eines Decks und um das Verständnis der Funktion des Decks.

Es ist ein einfacher solarer Durchlauferhitzer gemäß Bild 26 zu realisieren. Eine Pumpe fördert durch einen horizontalen Kollektor von 1 m^2 einen konstanten Volumenstrom genau dann, wenn die Globalstrahlung $200 \text{ W/m}^2 = 720 \text{ kJ/hm}^2$ übersteigt. Zu ermitteln sind die mittleren Nutzungsgrade für jeden Monat und für das gesamte Jahr.

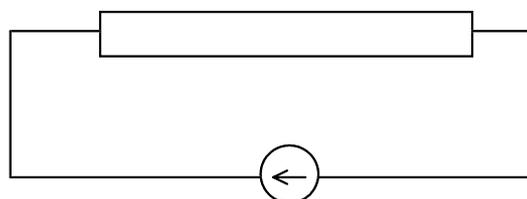


Bild 26: Einfacher solarer Durchlauferhitzer

Gegeben ist ebenfalls eine Wetterdatei mit Namen dd3.wet, in der sich u. a. stündliche Werte für Globalstrahlung und Außentemperatur befinden. Zunächst wird ein Informationsflußbild erstellt, das heißt, Bild 26 wird durch die externen Types und die Informationsflüsse sowie um Type und Unitnummern ergänzt.

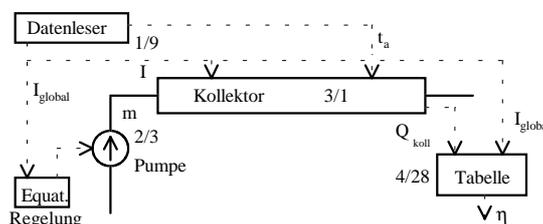


Bild 27: Informationsflußbild

An Komponenten werden ein Datenleser, eine Pumpe, ein Kollektor und eine Ausgabeeinheit, in diesem Fall eine Ergebnistabelle, benötigt. Weitere Festlegungen sind für die Definition der externen Dateien, der Simulationszeit, der Errechnung des Regelsignals und dem Simulationsende notwendig. Entsprechend den Parameter-, Input- und Outputbelegungen ergibt sich ein Deck gemäß Bild 28.

```

* Einfacher solarer Durchlauferhitzer

Assign bsp1.out      6      Zuweisung der Ausgabedatei
Assign dd3.wet      10      Zuweisung der Wetterdatei

Simulations 0 8760 1      Festlegung von Simulationsbeginn,
                           -ende und -zeitschritt

Unit 1 Type 9 Datenleser
Parameters 9
7 1 -5 1 0 -6 1 0 10      Type-Nummer
                           Bestimmung des Regelsignals.

Equations 1
ga = GT([1,6],720)        Der Output [1,6] wird mit der Vorgabe
                           verglichen und die Regelvariable ga
                           bei Übersteigen auf 1, sonst auf 0, gesetzt

Unit 2 Type 3 Pumpe
Parameters 1
30
Inputs 3
0,0 0,0 ga              Input als Variable ga gemäß Gleichung
20 30 0                 Input (1. Input Pumpe = Eintrittstemperatur) konst.
                           Konstantwert dieses Inputs (20 °C)

Unit 3 Type 1 Kollektor
Parameters 6
1 4.19 0.83 12.6 0.04 0.15      Parameterliste (Belegung gemäß Katalog)
                               3. Parameter Kollektor (opt. Wirkungsgrad = 0,83)
Inputs 6
2,1 2,2 1,7 1,6 1,5 0,0      5. Input Kollektor (Einstrahlung)
20 0 -6.2 0 0 60            vom Unit 1 den 5. Output beziehen
                               Anfangswert ist 0

Unit 4 Type 28 Ergebnistabelle
Labels 1
ETA%                        Spaltenüberschrift in der Tabelle
Inputs 2
3,3 1,6                      Zuweisung der Daten in die Datei Nummer 6
                               nach Assign-Anweisung ist dies bsp1.out
Parameters 12
-1 0 8760 6 2
0 0 2 -1 100 1 -4           Ende des Decks

End

```

Bild 28: Beispieldeck mit Erläuterungen

2.3.3 Konvergenz und Genauigkeit

Der Zustand der Anlage bei jedem Zeitschritt wird mittels trivialer Iteration bestimmt. Dadurch ergeben sich bei komplizierten Schaltungen schnell beträchtliche Konvergenzprobleme, die dem TRNSYS-Nutzer große Schwierigkeiten bereiten. Die Iterationszahl steigt und damit auch die Rechenzeit. Zum Teil wird keine Konvergenz gefunden, was einige reale Betriebsführungen unberechenbar machen kann, beispielsweise Kollektoren mit sehr hohen spezifischen Masseströmen.

Unlösbare Konvergenzprobleme entstehen vorrangig durch Regelalgorithmen und Anlagenkennlinien, die Sprünge aufweisen. Ein typischer Fall ist der Versuch, die Fensterverschattung un stetig nach der Raumtemperatur zu regeln. Dabei wird in einer Formel definiert, daß bei Überschreiten einer bestimmten Raumtemperatur die Verschattung zu schließen und diese bei Unterschreiten wieder zu öffnen ist.

Es kann hierbei der Fall eintreten, daß die Raumtemperatur durch das Schließen der Verschattung noch während des Zeitschrittes unter die Grenztemperatur fällt, was wieder zum Öffnen der Verschattung führt. Wenn es bei geöffneter Verschattung aber nicht zum Unterschreiten des Grenzwertes kommt, errechnet TRNSYS erneut einen Schließbefehl. Dieser Rechenhergang wird sich so oft wiederholen, bis die maximale Iterationszahl erreicht und die Berechnung wegen Inkonvergenz abgebrochen wird.

Das Problem läßt sich dadurch lösen, daß der Öffnungszustand der Verschattung stetig in Abhängigkeit von der Raumtemperatur in einem bestimmten Bereich geregelt wird. Eine zweite Möglichkeit bietet der Einsatz eines Regler-Type, bei dem eine Hysterese vereinbart wird. Bei ungünstigen Hysterese-Werten kann es aber dennoch zur Inkonvergenz kommen.

Bei der Programmierung neuer TRNSYS-Module ist darauf zu achten, daß die der Modellierung zugrundeliegenden empirischen Kennlinien oder auch physikalischen Modelle keine Unstetigkeiten in Form von Wertesprüngen aufweisen sollten. In einem derartigen Fall kann der Nachnutzer auch durch eine geschickte Deckerstellung Konvergenzprobleme nicht vermeiden.

Konvergenzprobleme treten auch dann auf, wenn sich Speichervorgänge beschreibende Größen innerhalb eines Zeitschritts zu stark ändern, zum Beispiel die Temperatur in einem Gebäude. Es empfiehlt sich dann, die Zeitschrittweite zu verringern und der langsamen Konvergenz dadurch Rechnung zu tragen, daß die zulässige Iterationsanzahl durch eine LIMITS-Anweisung erhöht wird.

Die durch das Berechnungsverfahren der trivialen Iteration bedingte, langsame Konvergenz soll anhand eines Beispiels demonstriert werden. Ausgangspunkt ist der solare Durchlauferhitzer gemäß Bild 26. Dieser soll dahingehend verbessert werden, daß eine bestimmte Austrittstemperatur des Wassers gewünscht wird. Realisierbar wird dies durch eine proportionale Regelung des Volumenstroms gemäß folgender Formel:

$$m_{\text{koll}} = m_{\text{max}} \cdot (t_{\text{ist}} - t_{\text{soll}}) / \Delta t_{\text{max}} \quad 0 < m_{\text{koll}} < m_{\text{max}}$$

Die Iteration beginnt mit der Kollektorausstrittstemperatur, bei welcher der vorangegangene Zeitschritt geendet hatte. Daraus wird ein Massestrom berechnet, der dem Kollektor zugeführt wird, wonach der Kollektor eine neue Austrittstemperatur bestimmt, die im zweiten

Iterationsschritt zur Berechnung eines neuen Massestroms führt. In Bild 29 ist das langsame Voranschreiten der Kollektortemperatur im Iterationsprozeß dargestellt.

Um ausgewählte Größen mit besonders langsamer Konvergenz mit weniger Iterationsschritten berechnen zu können, enthält TRNSYS ein Type "Konvergenzbeschleuniger", das den Iterationsvorgang beobachtet und ab dem 3. Iterationsschritt entsprechend dem Sekantenverfahren neue Werte in den Berechnungsvorgang einspeist.

In Bild 29 ist ebenfalls der Iterationsverlauf bei Einsatz des Konvergenzbeschleunigers dargestellt. Ohne Konvergenzbeschleuniger, also bei trivialer Iteration beträgt die Genauigkeit nach 20 Iterationen 0,1 K und nach 34 Iterationen 0,01 K. Der Konvergenzbeschleuniger stellt bereits nach 5 Iterationen eine Genauigkeit von 0,001 K bereit

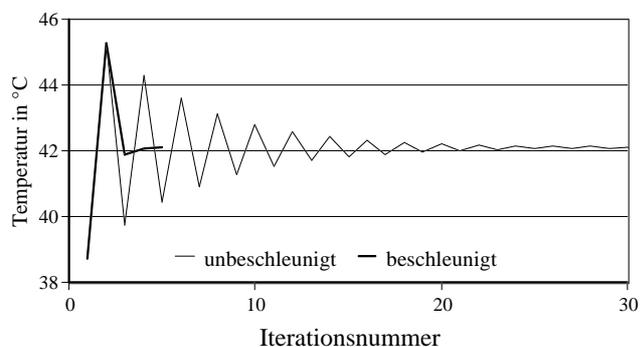


Bild 29: Konvergenzverhalten

Beispielrechnungen ergaben, daß der Konvergenzbeschleuniger auch Zeitschritte, die bei trivialer Iteration divergieren, zur Konvergenz bringen kann. Das dargestellte Beispiel ist ein typischer Fall dafür, da die Gleichung zur Berechnung des Volumenstroms bei geringen zulässigen Regelabweichungen Δt_{\max} unstetig im Sinne starker Knickpunkte wird. Die iterative Berechnung schwankt ohne Konvergenzbeschleuniger dann ständig zwischen den Berechnungen für vollen und für nicht vorhandenen Volumenstrom und kann den dazwischen liegenden Lösungswert nicht erreichen.

Einzigster Nachteil des Konvergenzbeschleunigers ist, daß jeweils nur eine Größe beschleunigt werden kann. Diesen Nachteil weist die "ACCELERATE"-Anweisung, die laut TRNSYS-Manual eine Beschleunigung nach dem gleichen mathematischen Prinzip zur Folge hat, nicht auf. Eine Beispielrechnung ergab allerdings, daß diese Anweisung keinen Einfluß auf den Iterationsverlauf nimmt.

Die bei der TRNSYS-Simulation angestrebte Genauigkeit wird durch eine "TOLERANCE"-Anweisung definiert. Diese beeinflusst vorrangig die Genauigkeit der Temperaturberechnung. Bei Systemen, die sehr geringe Temperaturspreizungen aufweisen, wie zum Beispiel Sorptionskältemaschinen, dürfen die Temperaturabweichungen daher nur gering sein, um eine ausreichende Genauigkeit der Berechnung der Wärmeströme sicherzustellen.

2.3.4 Die Modellierung des Gebäudes

Kompliziertester und oft rechenzeitintensivster Gegenstand von Simulationen ist zumeist das Gebäude. Jede Modellierung kann hierbei nur eine Näherung unterschiedlichen Grades an die Realität sein.

Die Nutzungsbedingungen des Gebäudes können ausgesprochen variabel gestaltet werden. So sind innere Quellen, das Heizungsregime, Lüftung, Wärmeübergangskoeffizienten, Grenztemperaturen und Einstrahlungen als Zeitschwingungen oder als beliebig von anderen Größen abhängig definierbar. Einen Überblick über Art und Vielfalt der ein Gebäude beschreibenden Größen gibt Bild 30.

Ausgehend von den bisher gültigen Einschränkungen der Rechentechnik ist der Rechenumfang des Gebäudes in der Standardversion sehr begrenzt. So sind nur 10 Zonen und 20 unterschiedliche Wände zulässig. Wer einige FORTRAN-Kenntnisse besitzt, kann diese Parameter relativ problemlos erhöhen.

Wesentlich eingeschränkt wird die Genauigkeit der Gebäudesimulation durch das nichtgeometrische Modell. Es werden keine Einstrahlzahlen berechnet, was vor allem die Berechnung von Wandheizflächen mit Fehlern behaftet. Zudem wird die Sonneneinstrahlung durch die Fenster gleichmäßig auf alle Wände der Zone verteilt. Dies ist in beschränktem Maße durch die Korrektur der Absorptionswerte ausgleichbar. Da nur ein Luftknoten je Zone existiert, kann die vertikale Temperaturverteilung nicht berücksichtigt werden.

Relativ genau werden die Fenster modelliert. So wird die Winkelabhängigkeit von Absorption und Reflexion abhängig von Scheibenzahl und Brechzahl des Glases berücksichtigt.

Die größten Probleme bereitet gegenwärtig die Stabilität der Transferfunktion bei dicken Wänden. Gegenwärtig ist nur eine begrenzte Anzahl von Schritten gestattet, was zum Beispiel die exakte Berechnung von Ziegelwänden über 40 cm Dicke unmöglich macht. Es ist möglich, im Quellcode die zulässige Anzahl zu erhöhen, wobei aber die Berechnung immer ungenauer wird. Es können nicht erkennbare Fehler auftreten.

2.3.5 Der TRNSYS-Variator

Im Rahmen der vorliegenden Arbeit wurden mehrere neue TRNSYS-Types erstellt. Diese dienen der Lösung spezieller energietechnischer Probleme und werden daher in den entsprechenden Abschnitten der Arbeit beschrieben. Lediglich das Variationsproblem ist allgemeiner Natur und wird daher im folgenden hinsichtlich Problemstellung und Lösung dargelegt.

Bisher konnte mit TRNSYS in einem Simulationslauf nur eine Anlagenkonfiguration untersucht werden. Damit ist es nur ungenügend für Design und Optimierung geeignet. Zur Bestimmung eines optimalen Parameters wird TRNSYS bislang mit einem bestimmten Deck gestartet. Nach Beenden der Simulation wird dieser Parameter im Deck geändert und das Programm erneut gestartet. Bei praktischen Aufgabenstellungen sind oft mehrere Größen zu variieren, was zu einem immensen Arbeitsaufwand führt. Sehr viel günstiger wäre die automatische Abarbeitung eines solchen Prozesses. Dann könnten zum Beispiel Simulationsvariationen über Nacht vom Computer berechnet werden. Im Rahmen dieser Arbeit wurde diese Programmieraufgabe bei einem Spezialisten in Auftrag gegeben /46/.

Ursprünglich bestand die Vorstellung, ein übergeordnetes Programm zu entwickeln, welches jeweils in das Deck eingreift, einzelne Größen ändert und neu startet. Es wurde jedoch eine sehr viel elegantere Lösung gefunden, indem die Variation durch ein neu geschaffenes TRNSYS-TYPE erfolgt. Dieses kann in der vorliegenden Form bis zu zehn Parameter oder konstante Inputs variieren. Es mußten einige andere Types geändert werden, in denen die Initialisierung durch DATA-Anweisungen erfolgte.

Der Variator wird wie jede andere Komponente kompiliert und gelinkt. Die Wetterdaten dürfen nur in einer Datei vorliegen, da der Datenleser bei der Initialisierung nur auf die gerade benutzte Wetterdatei zurücksetzen kann. Die Variation von Ausgabeeinheiten ist nicht sinnvoll und auch nicht möglich. Die Ausgabe der Ergebnisse erfolgt in eine Datei in der Reihenfolge der Simulationsläufe. Dabei geht der jeweils letzte Zeitschritt jeder Simulation verloren. Der Fehler bei einer Ganzjahressimulation ist aber gering. Im folgenden ist die Parameterbelegung dargestellt. Outputs sind nicht vorhanden.

Par- Nr.	Belegung
·	·
5*i+1	UNIT Nummer der (i+1) ten zu variierenden Größe
5*i+2	Position der zu variierenden Größe in der Liste
	pos. Vorzeichen: ein Parameter ist zu variieren
	neg. Vorzeichen ein konstanter Input ist zu variieren
5*i+3	Startwert
5*i+4	Endwert
5*i+5	Schrittweite
·	·

Die Laufvariable i liegt zwischen 0 und 9. Damit können bis zu 10 Größen variiert werden, müssen aber nicht. Vor dem Start der Simulation ist zu bedenken, daß die Variationsgesamtzahl mit der Variationsbreite stark ansteigt, also nach Möglichkeit nicht annähernd ausgeschöpft werden sollte. So dauert, selbst wenn jede von 10 Größen nur dreimal variiert wird und jede Simulation 50 Sekunden dauert, die gesamte Variation mehr als einen Monat. Ebenso muß die Größe der Ausgabedatei vorher kalkuliert werden. Sie kann schnell eine festplattensprengende Größe erreichen. Tritt dies während der Variation ein, stürzt der Rechner ab und die bereits ermittelten Ergebnisse dieser Variation gehen verloren.

Bei einigen Untersuchungen, speziell saisonaler Speicher, müssen Anfangs- und Endzustände abgeglichen werden. Es ist dann sinnvoll, identische Variationen zu wiederholen. Zu diesem Zweck wird die Berechnungshäufigkeit als konstanter Input definiert. Weitere Inputs existieren nicht.

Für die automatische Parametervariation können alternativ zum Variator auch Programme in speziellen Programmiersprachen verwendet werden. Eine Möglichkeit ist der Einsatz von "Visual Basic für Applikationen" (VBA), das zum Lieferumfang von MS-Excel gehört. Ein vom Nutzer erstelltes Programm liest die Parameter ein, erstellt das TRNSYS-Deck, startet die Simulation und liest die Ergebnisse aus.

Vorteile dieses Verfahrens sind die Simulationssteuerung vom Excel und die Möglichkeit der Optimierung. Nachteile bestehen in der Notwendigkeit des Erlernens einer weiteren Programmiersprache (VBA) und des Einsatzes windows-kompatibler TRNSYS-Programme. Letzteres schränkt den möglichen Simulationsumfang ein und verlängert die Rechenzeit.