Control Structure Design in Process Control Systems

Johansson, Karl Henrik; Hägglund, Tore

1999

ISSN 0280–5316
ISRN LUTFD2/TFRT--7585--SE

Control Structure Design in Process Control Systems

Karl Henrik Johansson
Tore Hägglund

Department of Automatic Control
Lund Institute of Technology
September 1999

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
INTERNAL REPORT

Date of issue
September 1999

Document Number
ISRN LUTFD2/TFRT--7585--SE

Author(s)
Karl Henrik Johansson and Tore Hägglund

Supervisor

Sponsoring organisation
Swedish Foundation for Strategic Research

Title and subtitle
Control Structure Design in Process Control Systems

Abstract

The control configuration problem is discussed. A new algorithm is outlined which gives suggestion on feedback and feedforward control structure given a SISO control loop and a number of extra measurements. The algorithm is based on simple experiments and leads to a model of the process as a directed graph. The method is illustrated on a number of common industrial control configurations. Relations to Kalman decomposition are also presented.

Key words
Control configuration; Multivariable control; Process control

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
19

Recipient's notes

Security classification

1. Introduction

Autonomy in process control systems is increasing in importance as a result of growing complexity of industrial systems [Antsaklis *et al.*, 1991; Åström, 1991]. The configuration of the controllers is an important factor, although today it is often not considered as a crucial variable when process designs are updated. Control structures in industry have traditionally evolved through years of experience. Rapid development of sensor and computer technology has, however, given new possibilities to make major structural changes in many process designs. This has led to an increasing need for automatic or semi-automatic control structure design tools. Finding a suitable structure or choosing between different structures are in general difficult problems. Even though these type of problems can be regarded as multivariable control problems, little of the activity in multivariable control the last three decades has been devoted to these problems, see discussions in [Shinskey, 1981; Skogestad and Postlethwaite, 1996].

The main contribution of this report is an algorithm for control structure design. The algorithm consists of a sequence of experiments that lead to a structural model of the plant, which automatically suggests a control configuration. No prior information about the process is needed. The particular setup is discussed when a SISO control loop is given and a number of extra measurements are available. It is shown that a graph is a natural model for such a system. The graph tells the role each measurement should play in the controller. The problem statement includes many interesting industrial cases. Some of them are discussed as examples in the report and it is shown that in these cases the algorithm leads to the same control structure as the ones used in practice. Graph theory is used in various areas in control engineering. Directed graphs have, for example, been used in the study of large scale systems and decentralized control problems since the early seventies [Reinschke, 1988; Šiljak, 1991]. The modeling we study here is also related to qualitative reasoning [Bobrow, 1985; Kuipers, 1985], as we are not primarily concerned about detailed dynamical models but more qualitative properties such as causality. Supervision of process control systems is an example of an area where causal reasoning has been investigated [Montmain and Gentil, 1999].

The outline of the report is as follows. The definition of a process graph is given in Section 2 together with some other preliminaries. Section 3 presents an algorithm for identifying a process graph from some simple step experiments. Control structure design is discussed in Section 4, where a number of design rules based on the process graph are given. Section 5 lists some common industrial control configurations and how these are detected by the algorithm in this report. Process graph modeling is closely related to Kalman decomposition. This relation is formalized in Section 6. Some extensions and future work are discussed in Section 7 and the conclusions are given in Section 8. Pseudo-code for the algorithms in Section 3 is given in an appendix.

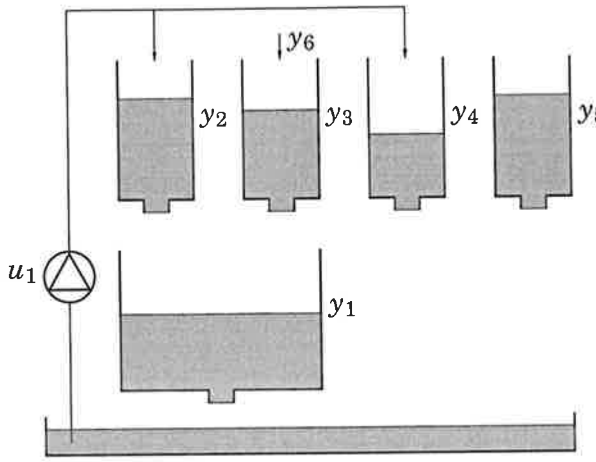

Figure 1 Water tank system with measured signals $y_1, \ldots, y_6$ and one control signal u_1. The objective is to control y_1.

2. Process Graph

Consider a multivariable control system with control signals $u_1, \ldots, u_m$, measured signals $y_1, \ldots, y_p$, and reference signals $r_1, \ldots, r_q$. Each reference signal r_k is associated to a measured signal y_k. It holds that $p \geq q$ and the interesting case here is when there is strict inequality. The control objective is loosely defined as to keep y_k as close to r_k as possible regardless of external disturbances, changing operating conditions, cross-couplings, unmodeled dynamics etc. We are interested in how to choose a good control configuration to solve this problem, but we will not discuss particular choices of control parameters.

It is illustrative for our purposes to model the process as a directed graph, where each node represents a measured signal and each edge a dynamical connection.

DEFINITION 1—PROCESS GRAPH
A *process graph* G is a directed graph $G = (V, E, W)$, where the sets $V = \{u_1 \ldots, u_m, y_1, \ldots, y_p\}$ and $E \subset V \times V$ are nodes and edges, respectively, and the map $W : E \mapsto D$ associates an input–output map to each edge. □

For example, for a process graph representing a linear time invariant system, W maps each edge to a finite-dimensional transfer function. If each edge represents a time delay, we have instead $D = \{\exp(-sL) : L \geq 0\}$.

EXAMPLE 1—WATER TANK SYSTEM
Consider the water tank system in Figure 1, which consists of five tanks and a pump. The control objective is to keep the level y_1 close to the set-point r_1. The control signal u_1 is the input to the pump. The measurements $y_2, \ldots, y_5$ are levels, while y_6 is a flow. If we neglect the dynamics in the pump, time delays in pipes etc., and only consider the dynamics resulting from Bernoulli's equation, then the linearized dynamics between the pair of signals corresponding to the connected tanks can be represented by first-order transfer functions. The process graph for this model is shown in Figure 2, where the weighting W is suppressed. □

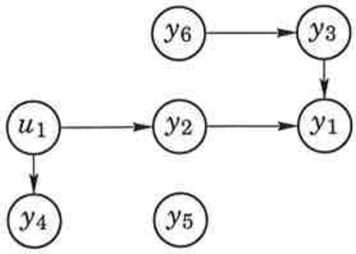

Figure 2 Process graph that represents the tank system in Figure 1.

Introduce succ(·) and pre(·), which map subsets of V to subsets of V, as

$$\begin{aligned}\text{succ}(U) &:= \{v \in V : (w,v) \in E, w \in U\}\\ \text{pre}(U) &:= \{v \in V : (v,w) \in E, w \in U\}.\end{aligned}$$

They hence denote the successors and the predecessors, respectively, for a set of nodes. Define $\text{succ}^k(\cdot)$ iteratively as $\text{succ}^0(U) = U$ and $\text{succ}^k(U) = \text{succ}\big(\text{succ}^{k-1}(U)\big)$ for $k \geq 1$. The map $\text{pre}^k(\cdot)$ is defined similarly. To emphasize the underlying graph G, we sometimes use a subscript as in succ_G and pre_G.

A *path* in a process graph is a sequence $\{v_i\}_{i=1}^k$, $v_i \in V$ and $k > 1$, such that $v_i \in \text{succ}(v_{i-1})$ for all $i = 2,\dots,k$. A process graph has a *cycle*, if there exists $v \in V$ and $k \geq 1$ such that $v \in \text{succ}^k(v)$. It is *acyclic* if there is no cycle. A process graph has a *parallel path*, if there exists two non-identical sequences $\{v_i\}_{i=1}^k$ and $\{w_i\}_{i=1}^\ell$, $k,\ell \geq 1$, such that $v_i \in \text{succ}(v_{i-1})$, $w_i \in \text{succ}(w_{i-1})$, $v_1 = w_1$, and $v_k = w_\ell$. A node w is *reachable* from v if there exists a path from v to w. Otherwise, the node is *unreachable*.

Throughout this report we focus on the case with a single scalar control loop with a few extra measurements, i.e., $m = q = 1$ and $p > 1$. Then, $V = \{u_1, y_1, \dots, y_p\}$. We assume that y_1 is reachable from u_1. Decompose $V \setminus \{u_1, y_1\}$ into four disjoint sets

$$V = \{u_1, y_1\} \cup V_{rr} \cup V_{ru} \cup V_{ur} \cup V_{uu},$$

where

$$\begin{aligned}V_{rr} &:= \{v \in V : v \text{ is reachable from } u_1,\ y_1 \text{ is reachable from } v\}\\ V_{ru} &:= \{v \in V : v \text{ is reachable from } u_1,\ y_1 \text{ is unreachable from } v\}\\ V_{ur} &:= \{v \in V : v \text{ is unreachable from } u_1,\ y_1 \text{ is reachable from } v\}\\ V_{uu} &:= \{v \in V : v \text{ is unreachable from } u_1,\ y_1 \text{ is unreachable from } v\}.\end{aligned}$$

Breaking up V like this is related to Kalman decomposition, as we will see in Section 6.

EXAMPLE 1—WATER TANK SYSTEM (CONT'D)
The process graph for the water tank system in Figure 2 has the paths $\{u_1, y_4\}$, $\{u_1, y_2, y_1\}$, and $\{y_6, y_3, y_1\}$. It has hence no cycles or parallel paths. For example, the pairs y_1, y_2, and y_4 are reachable from u_1, while y_5 is unreachable. The reachability structure is given by $V_{rr} = \{y_2\}$, $V_{ru} = \{y_4\}$, $V_{ur} = \{y_3, y_6\}$, and $V_{uu} = \{y_5\}$. □

3. Process Graph Identification

In this section an algorithm is derived to identify a process graph through a number of experiments. The obtained process graph is then used in the next section to derive a control structure. We limit the class of considered processes in order to give a transparent presentation. The process graph identification will in general not give full information about the system, in the sense that the true process graph is obtained. However, the intention is that after a number of experiments, the graph should be sufficiently accurate to suggest a suitable control structure.

The following assumptions are made:

A1 $V = \{u_1, y_1, \ldots, y_p\}$;

A2 $D = \{\exp(-sL) : L \geq 0\}$; and

A3 G is acyclic and has no parallel paths.

Hence, we consider process graphs with (A1) one control signal, (A2) dynamics given by time delays, and (A3) with no cycles or parallel paths. Relaxations of these assumptions are discussed in Section 7, but already here we point out that the dynamical restriction is not severe. It only emphasizes that the control structure algorithm relies on causality, and not really on any true dynamical properties of the plant. The time delay of a response can be interpreted as the settling time for a system with more general dynamics. The considered types of plants have up to tens of measurements. The described structuring may not be applicable for large scale systems due to combinatorial reasons.

Transient response experiments are performed in order to obtain the process graph of the system. In the report we consider step experiments, but other excitation signals, such as pulses and sinusoids, may be preferable in some cases. The experiments are done in open loop. We define a *v experiment* as a step change in $v \in V$, such that all $w \in \text{pre}(v)$ are unaffected. The step is assumed to be induced by an actuator not modeled by the process graph. It is hence assumed that each measurement can be perturbed by an external variable. The perturbation may, for instance, be caused by manually opening and closing a valve. For the tank system in Figure 1, a y_2 experiment may be done by externally adding some water to Tank 2.

The process graph identification is divided into three algorithms. The initial information about the process graph G is assumed to be $G_0 = (V, E_0, \cdot)$ with $E_0 = \{(u_1, y_1)\}$. Hence, we only know the set of measurements and that there is a dynamic connection between u_1 and y_1. The result of Algorithm i is given by the process graph G_i. The response times $T_v(k) \in \mathbb{R}^+ \cup \{\infty\}$ from a step experiment in $v \in V$ are collected in the set $T_v = \{T_v(k)\}_{k=1}^{p}$. The notation $T_v(w)$ is sometimes also used for the response time of $w \in V$.

The algorithms are presented in pseudo-code in Appendix. Here we illustrate them through flow charts. For simplicity, the flow charts do not describe how the weighting W is obtained from the response time measurements T_v. Algorithm 1 consists of a u_1 experiment, i.e., an open-loop step response in the control signal. Algorithm 2 consists of $y_k \in V_{ru} \cup V_{rr}$ experiments, i.e., step experiments in all signals corresponding to the nodes

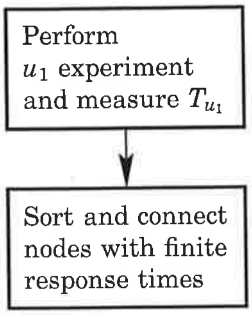

Figure 3 Flow chart illustrating Algorithm 1.

in $V_{ru} \cup V_{rr}$, obtained from Algorithm 1. Algorithm 3 finally consists of $y_k \in V_{ur} \cup V_{uu}$ experiments.

Algorithm 1 is illustrated by the simple flow chart in Figure 3. The algorithm connects each node that has a finite response time to a single path originating in u_1. The algorithm leads to the process graph $G_1 = (V, E_1, W_1)$ with $E_1 = \{(\text{succ}^{k-1}(u_1), \text{succ}^k(u_1))\}_{k=1}^n$ for some $n \in \{2, \ldots, p\}$. It holds that $y_1 = \text{succ}^\ell(u_1)$ for some $\ell \in \{2, \ldots, p\}$.

Algorithm 2 splits the graph consisting of a single path obtained from Algorithm 1 into a tree. As is shown in Figure 4, the algorithm has a loop, which steps through each node in the path and performs experiments. If y_1 is not affected by the experiment, the node is removed from the path between u_1 and y_1. The result G_2 of Algorithm 2 is a tree with root u_1. All nodes in the connected part of G_2 are reachable from u_1, but y_1 may only be reachable from some of them.

Algorithm 3 connects V_{ur} to $V_{rr} \cup V_{ru}$ and results in the final estimate G_3 of the true process graph G. The flow chart in Figure 5 presents the algorithm. The notation C is used for the connected component of the graph and $U \subset V_{ur} \cup V_{uu}$ for the remaining nodes. If a $v \in U$ experiment results in a response in y_1, then the resulting graph with a single path from v to y_1 is sorted and hooked on to the existing graph. Otherwise v belongs to V_{uu} and should not be connected.

We illustrate Algorithms 1–3 on the water tank system. The weightings are not given in order to simplify the presentation.

EXAMPLE 1—WATER TANK SYSTEM (CONT'D)
Consider the water tank system again and assume that the only information available is given by the graph G_0, which is shown as the top left process graph in Figure 6. Algorithm 1 gives the process graph G_1 in the top right graph in the figure, under the assumption that the response in y_4 is faster than the response in y_2. Note that G_1 is not a subgraph of the true graph G in Figure 2. Based on G_1, Algorithm 2 suggests a y_4 experiment followed by a y_2 experiment. The step perturbation in y_4 gives the process graph in the bottom left diagram in Figure 6. The y_2 experiment does not change the configuration. Algorithm 3 suggests experiments in y_3, y_5, and y_6. The bottom right graph shows the result after the first two experiments and Figure 7 shows the final result G_3. Note that in this particular example, the algorithms converged to the true process graph. □

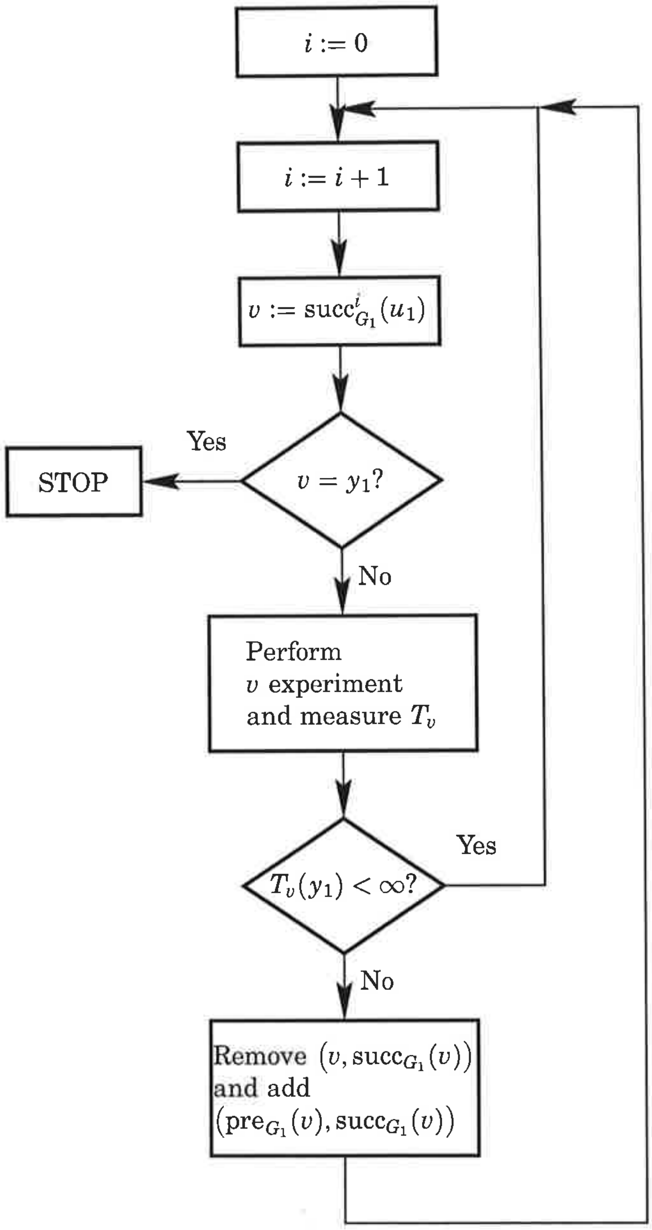

Figure 4 Flow chart illustrating Algorithm 2.

The main contribution of Algorithms 1–3 is to partition V into the subsets V_{rr}, V_{ru}, V_{ur}, and V_{uu}. The u_1 experiment in Algorithm 1 gives the nodes that belong to $V_{rr} \cup V_{ru}$. From this experiment it is, however, in general not possible to decide if y_1 is reachable or not from the nodes in $V_{rr} \cup V_{ru}$. This is done by the $y_k \in V_{rr} \cup V_{ru}$ experiments in Algorithm 2. All measurements in $V_{rr} \cup V_{ru}$ are perturbed and it is thus simple to separate the nodes in V_{ru} from the nodes in V_{rr}. The nodes that were not reachable from u_1 belong to $V_{ur} \cup V_{uu}$ and are perturbed in Algorithm 3. The $y_k \in V_{ur} \cup V_{uu}$ experiments tell if y_1 is reachable from these nodes or not, i.e., if a particular node belongs to V_{ur} or V_{uu}. Note that the process graph resulting from the algorithms in general differs from G. However, the decompositions of G_3 and G into sets V_{rr}, V_{ru}, V_{ur}, and V_{uu} are identical. In the next section we

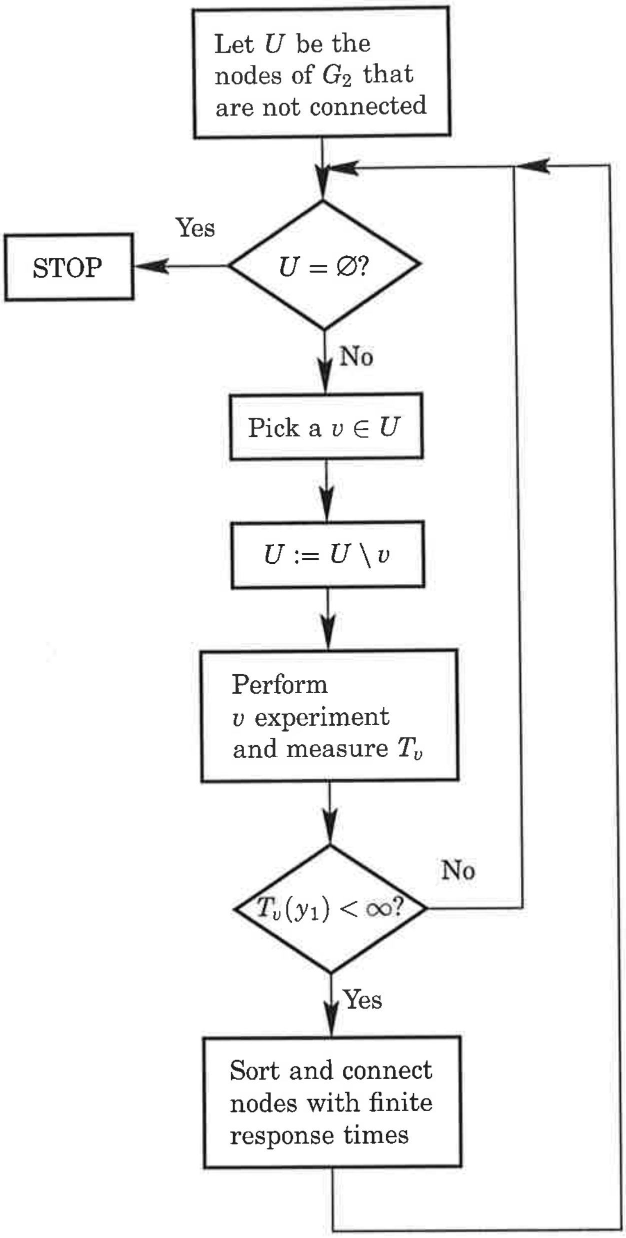

Figure 5 Flow chart illustrating Algorithm 3.

will see that the decomposition is crucial for the control structure design.

4. Control Structure Design

From the process graph it is possible to draw conclusions about a suitable control configuration. We discuss next how feedforward and cascade control structures can be found in the process graph.

The process graph in Figure 8 is a feedforward prototype. Here $V_{ur} = \{y_2\}$ and $V_{rr} = V_{ru} = V_{uu} = \varnothing$. The measured variable y_2 affects the controlled variable y_1. The signal y_2 may be a measurable disturbance or a variable that is related to a disturbance. The feedforward of the signal

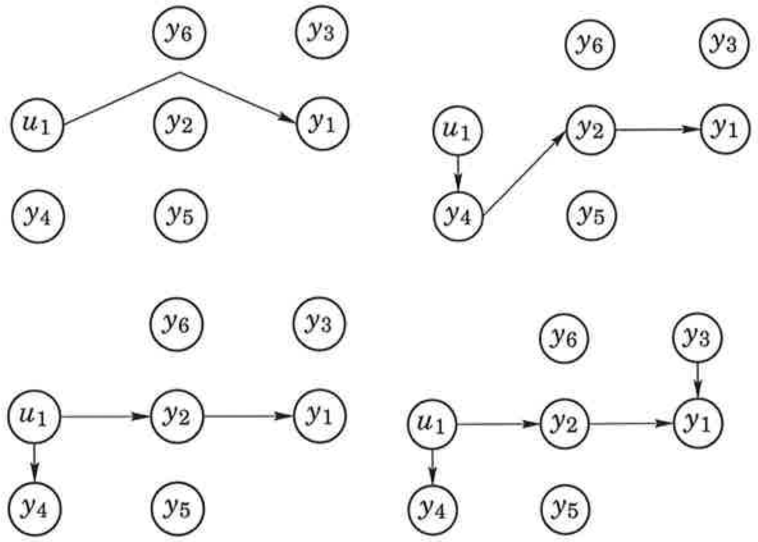

Figure 6 Process graph identification for the water tank system. The top left graph is the initial graph G_0. The top right is G_1 resulting from Algorithm 1, under the assumption that the response in y_4 is faster than the response in y_2. The bottom left graph is G_2 after Algorithm 2, which included experiments in y_4 and y_2. The process graph for the tank system after y_3 and y_5 experiments is bottom right. The final process graph G_3 is shown in Figure 7.

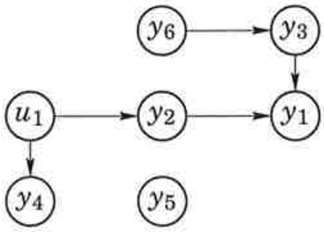

Figure 7 The process graph G_3 for the tank system after the final experiment, which is a y_6 experiment. In this example, the proposed algorithms give a process graph G_3 that is equal to the true graph G.

may improve the attenuation of this particular disturbance in the control loop. A closed-loop system with feedforward is illustrated in Figure 8. The block C_1 represents the SISO feedback controller, while the feedforward filter is denoted C_2.

A process graph prototype for cascade control is shown in Figure 9. Here $V_{rr} = \{y_2\}$ and $V_{ur} = V_{ru} = V_{uu} = \emptyset$. In this case the measured signal y_2 is responding to control actions faster than y_1. Therefore, it may be suitable to introduce an inner control loop based on tight control of y_2. Cascade control improves the performance considerably if there is an unmeasurable disturbance entering the system prior to y_2 and the y_1 response is much slower than the y_2 response. A cascade control loop is shown in Figure 9. The controller in the inner loop is denoted C_2 and the controller in the outer loop C_1. The objective is to regulate y_1. In many cases in practice, C_2 is a proportional controller. If there is only one node in the path between u_1 and y_1 that is used in feedback, cascade control is the commonly used term for the control structure. When there are two or more measurements fed back to the controller, we simply say feedback control. This case corresponds to

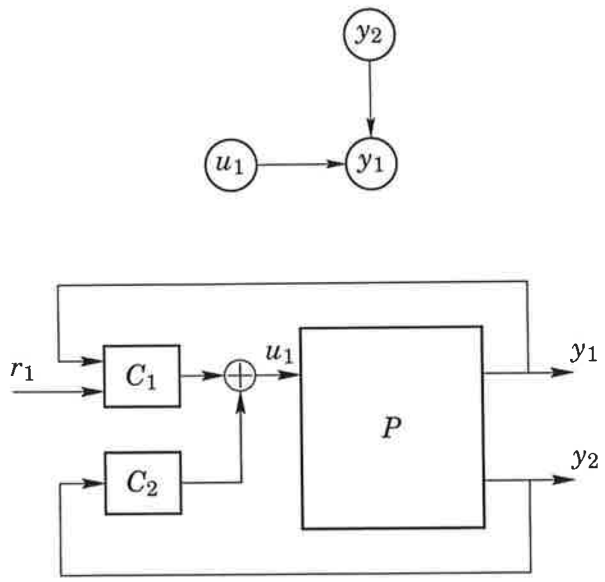

Figure 8 Prototype process graph suggesting feedforward control, together with a feedforward control structure.

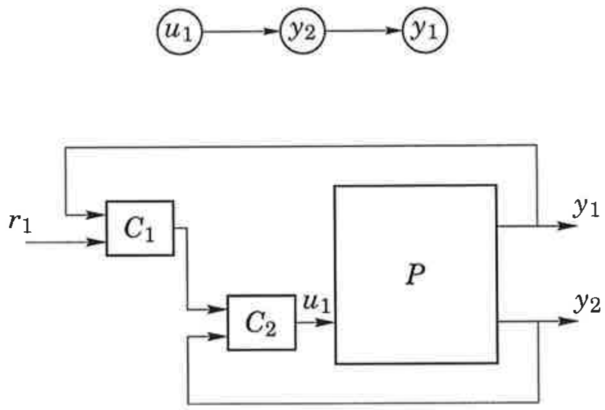

Figure 9 Prototype process graph suggesting cascade (feedback) control, together with a cascade control structure.

a control law based on (partial) state feedback.

By generalizing the conclusions from the previous three-node prototype graphs, we get the following control structure design rules based on the partition $V = \{u_1, y_1\} \cup V_{rr} \cup V_{ru} \cup V_{ur} \cup V_{uu}$:

- Measurements in V_{rr} may be used for feedback (cascade) control;
- Measurements in V_{ur} may be used for feedforward control; and
- Measurements in V_{ru} and V_{uu} should not be used for control of y_1.

There exist exceptions from these design rules. For example, there are cases when a measurement in V_{ru} is useful for feedback; for example, a sensor may have individual states from which y_1 is unreachable, but still these states reflect unmeasurable states in the process, which are useful for control of y_1. Another example when the rules should not be strictly followed is if there are redundant measurements or measurements

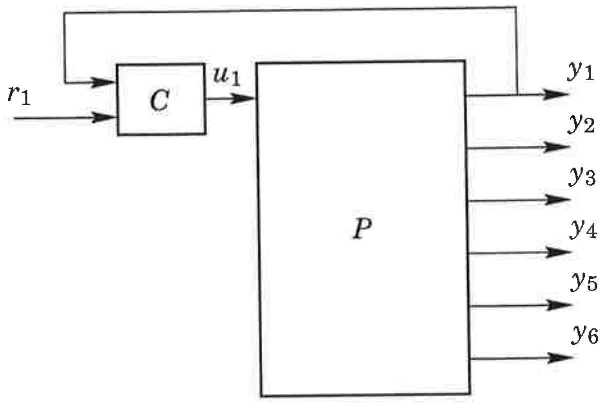

Figure 10 Existing control structure for the water tank system.

related by fast dynamics; then it might be sufficient to use only one of them. Further practical considerations are discussed in Section 7.

EXAMPLE 1—WATER TANK SYSTEM (CONT'D)
Assume that the present control structure for the water tank problem is as given in Figure 10. The question is if the control performance can be enhanced by using the measurements $y_2, \ldots, y_6$. The process graph identification in Section 3 gave the process in Figure 7 and $V_{rr} = \{y_2\}$, $V_{ru} = \{y_4\}$, $V_{ur} = \{y_3, y_6\}$, and $V_{uu} = \{y_5\}$. Following the control structure design rule, we have that y_2 may be used for feedback control, while y_3 and y_6 may be used for feedforward control. The other measurements should be neglected. This control structure is natural. The feedback control may be particular useful if there are (unmodeled) disturbances entering Tank 2. If y_6 is the only disturbance entering Tank 3 and if an accurate model of that tank is available, it is sufficient to feedforward only y_6. There are other cases when it is preferable to feedforward y_3 instead. □

5. Industrial Examples

In this section we illustrate the control structure design on three control problems that are common in process industry.

EXAMPLE 2—CONTROL OF A HEAT EXCHANGER
The top diagram in Figure 11 shows a process diagram for control of a heat exchanger. The control objective is to control the temperature on the secondary side (y_1) using the inlet valve on the primary side (u_1). There are often two additional measurement signals available: the flow on the primary side (y_2) and the flow on the secondary side (y_3). Following the algorithms in Section 3, it is easy to see that from open-loop steps experiments in u_1, y_2, and y_3, the resulting process graph is as given by the middle graph in Figure 11. For example, a change in u_1 results in a response in y_2, but not in y_3. The process graph suggests that the flow on the primary side should be used in feedback (cascade) and the flow on the secondary side in feedforward. The configuration is shown in the bottom

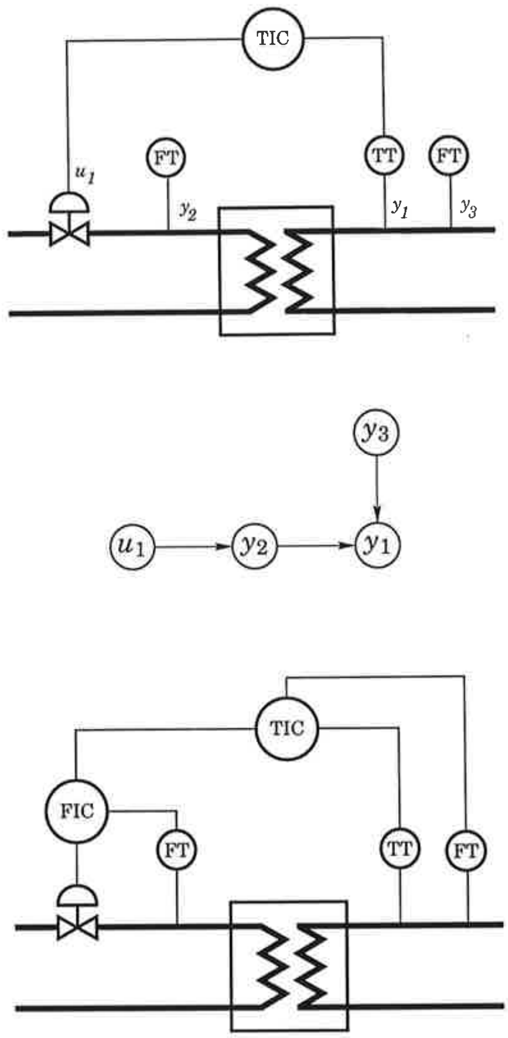

Figure 11 Control of a heat exchanger. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

diagram in the figure. This is the control configuration often used in practice for control of a heat exchanger. □

EXAMPLE 3—CONCENTRATION CONTROL

Figure 12 shows a process diagram for a concentration control loop. The control objective is to control the concentration of the blend (y_1) using the control valve on one of the two tubes (u_1). There is one additional measurement signal, the concentration of the media in the other tube (y_2). Process graph identification gives the middle graph in the figure. A change in u_1 will not result in any response in y_2, but a change in y_2 will result in a response in y_1. The design rules suggest that y_2 should be used in a feedforward connection, as shown in Figure 12. This is, of course, the natural configuration for this control problem. □

EXAMPLE 4—DRUM LEVEL CONTROL

Figure 13 gives a process diagram for level control of a drum boiler. The control objective is to control the level in the drum (y_1) using the feed-water

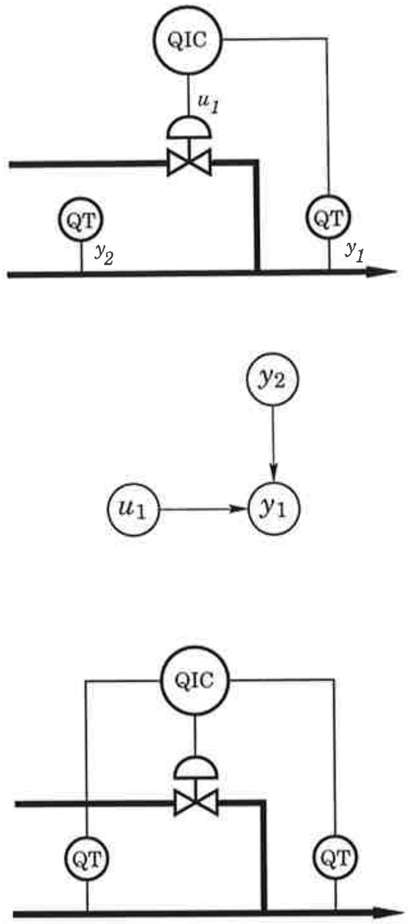

Figure 12 Concentration control. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

valve (u_1). There are two additional measurement signals, the feed-water flow (y_2) and the steam flow from the drum (y_3). The process graph is the same as for the heat exchanger problem, as illustrated by the middle diagram in Figure 13. A change in u_1 results in responses in y_1 and y_2, but not in y_3. This together with experiments in y_2 and y_3 gives the process graph in the figure. The control structure design rules give that the feed-water flow y_2 should be used in feedback (cascade) and the steam flow y_3 should be used in feedforward, as given by the bottom diagram in Figure 13 showing one of the standard configurations for drum level control. □

6. Kalman Decomposition

Finding the process graph of a system is closely related to Kalman decomposition. We illustrate this by considering a particular class of process graphs. Consider a process graph $G = (V, E, W)$ with $V = \{u_1, y_1, \ldots, y_p\}$ and no cycles or parallel paths. Let $W(e_i)$ be a stable strictly proper first-order transfer function for all $e_i \in E$. The process graph then models a linear time-invariant system and can thus be represented in state-space

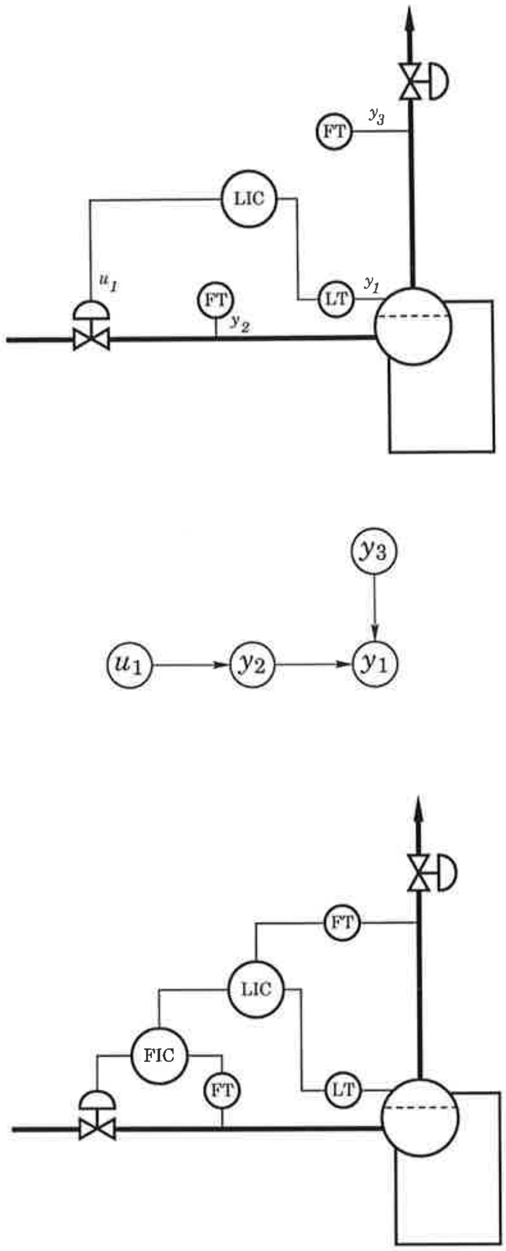

Figure 13 Drum level control. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

form. Consider the following Kalman decomposition of the state-space representation

$$\dot{x} = \begin{bmatrix} A_{11} & 0 & A_{13} & 0 \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & 0 & A_{33} & 0 \\ 0 & 0 & A_{43} & A_{44} \end{bmatrix} x + \begin{bmatrix} B_1 \\ B_2 \\ 0 \\ 0 \end{bmatrix} u_1 \tag{1}$$

$$y_1 = [\, C_1 \quad 0 \quad C_3 \quad 0 \,]\, x,$$

where

$$x = [\, x_{co}^T \quad x_{c\bar{o}}^T \quad x_{\bar{c}o}^T \quad x_{\bar{c}\bar{o}}^T \,]^T$$

and x_{co} represents the controllable and observable states, $x_{c\bar{o}}$ the controllable but unobservable states etc. [Kailath, 1980]. Since $W(e)$ are first-order transfer functions, each edge can straightforwardly be associated to a state x_i, which may be either controllable or uncontrollable and either observable or unobservable. It is easy to check the following four statements relating the process graph to the Kalman decomposition (1):

- $v \in V_{rr}$ if and only if there exist edges $(v,w),(w,v) \in E$ such that their associated states are controllable and observable;
- $v \in V_{ru}$ if and only if there exists an edge $(w,v) \in E$ such that the associated state is controllable and unobservable;
- $v \in V_{ur}$ if and only if there exists an edge $(v,w) \in E$ such that the associated state is uncontrollable and observable; and
- $v \in V_{uu}$ if and only if there exists an edge $(v,w) \in E$ or $(w,v) \in E$ such that the associated state is uncontrollable and unobservable, or there exist no nodes $(v,w),(w,v) \in E$.

It is possible to generalize the notion to incorporate other weightings W.

EXAMPLE 1—WATER TANK SYSTEM (CONT'D)
For the water tank process graph, we saw that $V_{rr} = \{y_2\}$, $V_{ru} = \{y_4\}$, $V_{ur} = \{y_3, y_6\}$, and $V_{uu} = \{y_5\}$. The state-space representation for the process graph with first-order transfer functions as weightings can easily be derived on the form (1), where $x_{co} = (x^1_{co}, x^2_{co})^T$ corresponds to the edges (u_1, y_2) and (y_2, y_1), $x_{c\bar{o}}$ corresponds to (u_1, y_4), and $x_{\bar{c}o} = (x^1_{\bar{c}o}, x^2_{\bar{c}o})^T$ to (y_6, y_3) and (y_3, y_1). No edge corresponds to an uncontrollable and unobservable state $x_{\bar{c}\bar{o}}$.

Note that the state-space system is not equal to the one obtained by deriving the equations for the physical system. For example, y_5 is obviously an uncontrollable and unobservable state, but this is not revealed by the process graph since no edge is associated to y_5. This comes from that the process graph does only model the dynamics *between* measurements. From the process graph we cannot judge that the measurement y_5 does not come from a static signal. □

Computational aspects are not discussed in this report. It is, however, interesting to notice the relation between the structuring of the plant and Kalman decomposition, because this relation suggests that powerful algorithms from subspace identification [Van Overschee and De Moor, 1996] could be used to obtain the process graph. Note, however, that the approach we have taken so far is to limit the computational efforts and build a framework that is just enough in order to capture the essentials to achieve a sufficiently good control structure.

7. Extensions

The methodology described in this report can be improved in a number of ways. Some of these extensions are presented here and include ongoing work.

Two Control Loops

The structuring problem becomes more involved as the number of controlled signals increases. In this report, we assumed the set of nodes $V = \{u_1, y_1, \ldots, y_p\}$ and that there is only one controlled variable y_1. Ongoing work includes extending the algorithm to two controlled variables y_1 and y_2 and the set $V = \{u_1, u_2, y_1, \ldots, y_p\}$. Even if there are no interaction between the control loops, the problem does not reduce to two separate design problems. The reason for this is that the new measurements may be dynamically linked to both control signals. Introducing a new measurement in one control loop may therefore give undesired interaction, although the original control loops were not coupled. Considering two control signals gives, however, the possibility to choose a multivariable feedback control structure, instead of just choosing between feedforward or scalar feedback control. The simplest extension to the method discussed in this report is to say that if there is interaction then decoupling should be used. A design rule for choosing decoupling in the spirit of previous sections is the following: if y_1 is reachable from u_2 and y_2 is reachable from u_1, then decoupling is suggested. Under the assumption that the decoupling is perfect, the control structuring algorithm can be applied on the decoupled process. Note, however, that even if there are no interaction between the decoupled control loops $\bar{u}_1$–y_1 and $\bar{u}_2$–y_2, there may still be interaction between the other measurements. Hence, adding feedforward and feedback control of these may introduce coupling between the two control loops. This coupling must not be neglected when choosing control structure.

Cycles and Parallel Paths

Throughout the report we have assumed that the process graph has no cycles or parallel paths. These assumptions are sometimes violated. Particularly cycles are natural in feedback systems: although the u_1–y_1 loop is assumed to be open, there might be other local loops that give cycles in the process graph. The assumptions on cycles and parallel paths can be made less restrictive by some careful considerations. Consider, for example, the upper process graph in Figure 14. If such a local cycle is found through (a modified version of) the identification, a meta node y_{23} should be introduced. In the reasoning about the process graph, this meta node then plays a similar role as a regular node. The interpretation is that y_2 and y_3 cannot be used independently in the control structure. If there are more loops involved in the cycle, all these should be replaced by a single meta node. Note that there are several new things to be considered about meta nodes, for example, to choose a measurement among the nodes defining the meta node. Meta nodes can be defined also for a process graph with a parallel path, as the one in the bottom graph in Figure 14.

Gain Scheduling

The control structure design algorithm suggests that measurements should be used for either feedforward or feedback. Another common use of external signals in industrial control systems is in gain scheduling, i.e., to let controller parameters depend on a measurement. A measurement should be used for gain scheduling if the *dynamics* of the process depends on it, in contrast to feedback and feedforward signals which affect the process output directly. This makes it harder to detect gain scheduling measurements.

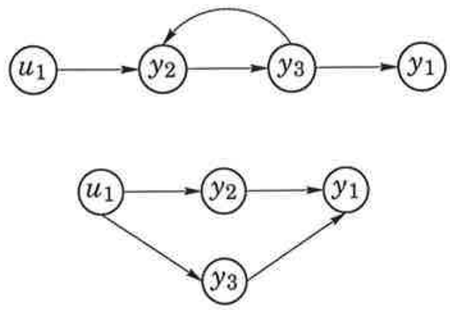

Figure 14 A process graph with a cycle and a process graph with a parallel path. Meta nodes are introduced to remove the cycle and the parallel path. For both graph the y_2 and y_3 nodes form the meta node.

Experiments that show that the process dynamics vary depending on the measurement could be followed by marking the node as a possible gain scheduling node. This require a modification of the definition of a process graph.

Initial Information

No prior information about the plant dynamics is assumed for the process graph identification. In practice, of course, some information about the dynamical coupling in the process and about disturbances and other uncertainties are available. This can (and should) be incorporated in the algorithm.

Knowledge about where in the process graph non-measurable disturbances enter is important for choosing measurements for feedback. A general principle for reducing the number of measurements is that if there are no dynamics or no disturbances entering the system between two measurements, then one of them can be neglected. One should try to find a measurement as close to the disturbance as possible; either outside the u_1–y_1 path and then use it for feedforward, or in the u_1–y_1 path and after the point where the disturbance enters and then use it for feedback.

Initial information about the process graph may come from process diagrams, process operator experiences, or earlier experiments. From these data an initial process graph can be drawn and the number of experiments in the process identification can be reduced. One special case is when there is a single new measurement available. It is then desirable to have automatic methods to find out if the signal is useful for feedforward or feedback control.

8. Conclusions

A control structure design algorithm was presented in this report. No prior information about the plant was required, but a process graph model was obtained through a number of simple step experiments. Most of the presentation was focused on a SIMO process, where one output represents the primary variable to control and the other outputs are candidates to possibly be included in the control structure. The process graph illustrates the causal relations in the process. This might be a pedagogical instrument to present control structures for process operators.

In the generic algorithm presented here, experiments are done for all nodes in the graph except for the controlled process output. The algorithm can easily be modified in order to limit the number of performed experiments. Note that if experiments are done for all nodes in the process graph, then it is better to evaluate the data altogether, instead of after each experiment.

An application of the control structure algorithm is in plant monitoring, where the ideas developed here can be used to online bring attention to unnecessary deterioration of plant performance caused by structural problems. For example, possible reduction of disturbances through feedforward may be found this way.

Ongoing work includes a few of the algorithm extensions discussed in Section 7. Implementations on industrial process control systems will be presented in future reports.

9. References

Antsaklis, P. J., K. M. Passino, and S. J. Wang (1991): "An introduction to autonomous control systems." *IEEE Control Systems Magazine*, **11:4**, pp. 5–13.

Åström, K. J. (1991): "Assessment of achievable performance of simple feedback loops." *International Journal of Adaptive Control and Signal Processing*, **5**, pp. 3–19.

Bobrow, D. G., Ed. (1985): *Qualitative Reasoning About Physical Systems.* MIT Press, Cambridge, MA.

Kailath, T. (1980): *Linear Systems.* Prentice-Hall, Inc, Englewood Cliffs, New Jersey.

Kuipers, B. (1985): *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* MIT Press, Cambridge, MA.

Montmain, J. and S. Gentil (1999): "Causal modeling for supervision." In *Proceedings of IMACS.*

Reinschke, K. J. (1988): *Multivariable Control—A Graph-Theoretic Approach*, vol. 108 of *Lecture Notes in Control and Information Science.* Springer, Berlin.

Shinskey, F. G. (1981): *Controlling Multivariable Processes.* Instrument Society of America, Research Triangle Park, NC.

Šiljak, D. D. (1991): *Decentralized Control of Complex Systems*, vol. 184 of *Mathematics in Science and Engineering.* Academic Press, Inc., San Diego, CA.

Skogestad, S. and I. Postlethwaite (1996): *Multivariable Feedback Control—Analysis and Design.* John Wiley & Sons.

Van Overschee, P. and B. De Moor (1996): *Subspace Identification for Linear Systems: Theory, Implementation, Applications.* Kluwer, Dordrecht.

Appendix

Algorithms 1–3, described in flow charts in Section 3, are presented in pseudo-code next.

ALGORITHM 1
Initial data: $G_0 = (V, E_0, \cdot)$, $V = \{u_1, y_1, \ldots, y_p\}$, $p > 1$, $E_0 = \{(u_1, y_1)\}$

> $E_1 := E_0$
> $W := V$
> Perform u_1 experiment
> Measure T_{u_1}
> $w := \arg\min_{v \in W}\{T_{u_1}(v)\}$
> <u>while</u> $T_{u_1}(w) < \infty$ <u>do</u>
> $E_1 := E_1 \cup (v, w)$
> $W_1(v, w) := T_{u_1}(w) - T_{u_1}(v)$
> $W := W \setminus v$
> $v := w$
> $w := \arg\min_{v \in W}\{T_{u_1}(v)\}$
> <u>end</u>

Result: $G_1 = (V, E_1, W_1)$ □

ALGORITHM 2
Initial data: $G_1 = (V, E_1, W_1)$

> $E_2 := E_1$
> $W_2 := W_1$
> $i := 1$
> $v := \mathrm{succ}_{G_1}(u_1)$
> <u>while</u> $v \neq y_1$ <u>do</u>
>   Perform v experiment
>   Measure T_v
>   <u>if</u> $T_v(y_1) = \infty$ <u>then</u>
> $E_2 := E_2 \setminus (v, \mathrm{succ}_{G_1}(v)) \cup (\mathrm{pre}_{G_1}(v), \mathrm{succ}_{G_1}(v))$
> $W_2(\mathrm{pre}_{G_1}(v), \mathrm{succ}_{G_1}(v)) := W_1(\mathrm{pre}_{G_1}(v), v) + W_1(v, \mathrm{succ}_{G_1}(v))$
>   <u>end</u>
> $i := i + 1$
> $v := \mathrm{succ}^i_{G_1}(u_1)$
> <u>end</u>

Result: $G_2 = (V, E_2, W_2)$ □

ALGORITHM 3
Initial data: $G_2 = (V, E_2, W_2)$

> $E_2 := E_1$
> $W_2 := W_1$
> $C := \{v \in V : \exists w \in V, (v, w) \in E_3 \text{ or } (w, v) \in E_3\}$
> $U := V \setminus C$
> <u>while</u> $U \neq \emptyset$ <u>do</u>

```
        Pick a $v \in U$
        Perform $v$ experiment
        Measure $T_v$
        if $T_v(y_1) < \infty$ then
            $W := V$
            $w := \arg\min_{\bar{w} \in W}\{T_v(\bar{w})\}$
            $\widehat{U} := U$
            $\widehat{v} := v$
            while $\widehat{v} \in \widehat{U}$ do
                $E_3 := E_3 \cup (\widehat{v}, w)$
                $W_3 := T_v(w) - T_v(\widehat{v})$
                $\widehat{U} := \widehat{U} \setminus w$
                $\widehat{W} := \widehat{W} \setminus w$
                $\widehat{v} := w$
                $w := \arg\min_{\bar{w} \in W}\{T_v(\bar{w})\}$
            end
        end
        $U := U \setminus v$
    end
```

Result: $G_3 = (V, E_3, W_3)$ □