CONTROL STRUCTURE DESIGN IN PROCESS CONTROL SYSTEMS

Karl Henrik Johansson

Department of EECS, University of California
Berkeley, CA 94720-1770, U.S.A.
johans@eecs.berkeley.edu

Tore Hägglund

Department of Automatic Control
Lund University, Sweden
tore@control.lth.se

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 some 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 few common industrial control problems.

1. INTRODUCTION

Autonomy in process control systems is increasing in importance as a result of growing complexity [1, 2]. The configuration of the controllers is crucial, although today it is often not considered as a 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 problems can be regarded as multivariable control problems, little of the activity in multivariable control the last three decades has been devoted to these issues [9, 11].

The main contribution of this paper 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 dynamics 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 paper 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 [8, 10]. The modeling we study here is also related to qualitative reasoning [3, 6], 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 [7].

The outline of the paper 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 transient response experiments. Control structure design is discussed in Section 4, where 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 proposed in the paper. Some extensions and future work are discussed in Section 6. An early version of this paper is available as [5].

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 $r_1, \ldots, r_q$, where $m \geq 1$ and $p \geq q \geq 1$. Each reference signal r_k is associated to the measured signal y_k. 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. In particular, we will suggest how to use $y_{q+1}, \ldots, y_p$ to improve closed-loop performance.

It is illustrative for our purposes to model the process as a directed graph, where each vertex represents either a control signal or 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 vertices and edges, respectively, and the weight function $W : E \to D = (0, \infty)$ associates a time to each edge.

The interpretation is that for all $e \in E$, the weight $W(e)$ represents the time delay between the two dynamically coupled signals connected by e. We could also allow more general dynamics imposed by D. For instance, letting D be the set of all first-order transfer functions with time delay. However, the control structure algorithm requires only information on causality, and not 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.

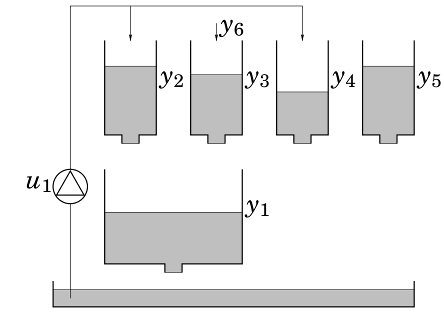

Fig. 1. Water tank system with measured signals $y_1, \ldots, y_6$ and one control signal u_1. The objective is to control y_1.

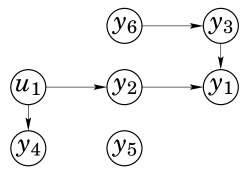

Fig. 2. Process graph that represents the tank system in Figure 1.

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. We may approximate the dynamics from the pump, the pipes, the tanks etc., by time delays. The process graph $G = (V, E, W)$ for this model is shown in Figure 2. Hence,

$$V = \{u_1, y_1, \ldots, y_6\}$$
$$E = \{(u_1, y_2), (u_1, y_4), (y_2, y_1), (y_3, y_1), (y_6, y_3)\}.$$

The weighting function W assigns a positive scalar to each edge (but is for simplicity suppressed in the figure).

Introduce succ$(\cdot)$ and pre$(\cdot)$, which map subsets of V to subsets of V, as

$$\text{succ}(U) := \{v \in V : \exists(w, v) \in E,\, w \in U\}$$
$$\text{pre}(U) := \{v \in V : \exists(v, w) \in E,\, w \in U\}.$$

They hence denote the successors and the predecessors, respectively, for a set of vertices. Define $\text{succ}^k(\cdot)$ iteratively as $\text{succ}^0(U) = U$ and $\text{succ}^k(U) = \text{succ}\left(\text{succ}^{k-1}(U)\right)$ for $k \geq 1$. The map $\text{pre}^k(\cdot)$ is defined similarly. A *path* in a process graph is a sequence $P = \langle v_1, \ldots, v_k \rangle$, $v_i \in V$ and $k > 1$, such that $v_i \in \text{succ}(v_{i-1})$ for all $i = 2, \ldots, k$. The weight of P is defined as $W(P) = \sum_{i=1}^{k-1} W(v_i, v_{i+1})$ (with abuse of notation). 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 $\langle v_1, \ldots, v_k \rangle$ and $\langle w_1, \ldots, w_\ell \rangle$, $k, \ell \geq 1$, 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 vertex w is *reachable* from v if there exists a path from v to w. Otherwise, the vertex is *unreachable*.

Example 1. (Cont'd). The paths for the water tank process graph in Figure 2 include $\langle u_1, y_4\rangle$, $\langle u_1, y_2, y_1\rangle$, and $\langle y_6, y_3, y_1\rangle$. It has no cycles or parallel paths. The vertices y_1, y_2, and y_4 are reachable from u_1, while y_5 is unreachable.

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. The process graph identification will in general not give full information about the system, in the sense that the true process graph will not necessarily be obtained. However, the intention is that after a number of experiments, the graph should be sufficiently accurate to suggest a suitable control structure.

Transient response experiments are performed in order to obtain the process graph of the system. In the paper we consider step experiments, but other excitation signals, such as pulses, 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.

We assume that it is possible to measure the response time $T_v(w) \in [0,\infty]$ from a step experiment in $v \in V$ to a signal $w \in V \setminus \{v\}$. The response times are collected as $T_v = \{T_v(w) : w \in V \setminus \{v\}\}$. We define $G_v = (V, E_v, W_v)$ as the process graph with vertices V, edges $E_v = \{(v,w) : T_v(w) < \infty\}$, and weight function $W_v(v,w) = T_v(w)$ for $(v,w) \in E_v$. Hence, G_v is a tree with root v and depth one. The distance graph$G_{\text{dist}} = (V, E_{\text{dist}}, W_{\text{dist}})$ is a process graph, computed from $\{G_v : v \in V\}$, such that $E_{\text{dist}} = \{(v,w) : \exists v,w \in V, (v,w) \in E_v\}$ and for $(v,w) \in E_{\text{dist}}$, $W_{\text{dist}}(v,w) = W_v(v,w)$. An process graph estimate $\hat{G}$ of the underlying graph G can be derived from G_{dist}. The process graph identification is summarized in the following algorithm.

```
for v ∈ V do
    Perform v experiment
    Measure T_v
    Compute G_v
od
Compute G_dist
Compute Ĝ
```

Let ShortestPath(G, v, w) denote the path with smallest weight between $v, w \in V$. Then, $\hat{G}$ is computed from G_{dist} by the following procedure.

```
Compute Ĝ = (V, Ê, Ŵ)
Ê := ∅
for v ∈ V do
    for w ∈ V \ {v} do
        P := ShortestPath(G_dist, v, w)
        if P ≠ ∅
          then if W_dist(P) > W_dist(v, w)
                  then Ê := Ê ∪ {(v, w)}
                       Ŵ(v, w) := W_dist(v, w)
                else E_dist := E_dist \ {(v, w)}
    od
od
```

The shortest path can easily be derived using Dijkstra's algorithm [4]. Under the assumption that G has no parallel paths or self-loops, the presented algorithm gives $\hat{G} = G$. We illustrate the algorithm on the water tank system. The weightings are not given in order to simplify the presentation.

Example 1. (Cont'd). Consider the water tank system again. The process graph identification algorithm leads to the process graphs $\{G_v : v \in \{u_1, y_1, \ldots, y_6\}\}$ shown as the first five graphs in Figure 3. These graphs result in the distance graph G_{dist} depicted as the graph in the lower right corner in Figure 3. Using Compute $\hat{G}$, we finally end up with a graph identical to the original graph G in Figure 2.

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 are suggested by the process graph. Now on 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, $V = \{u_1, y_1, \ldots, y_p\}$. We assume that y_1 is reachable from u_1. We also assume that G is acyclic and has no parallel paths. 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

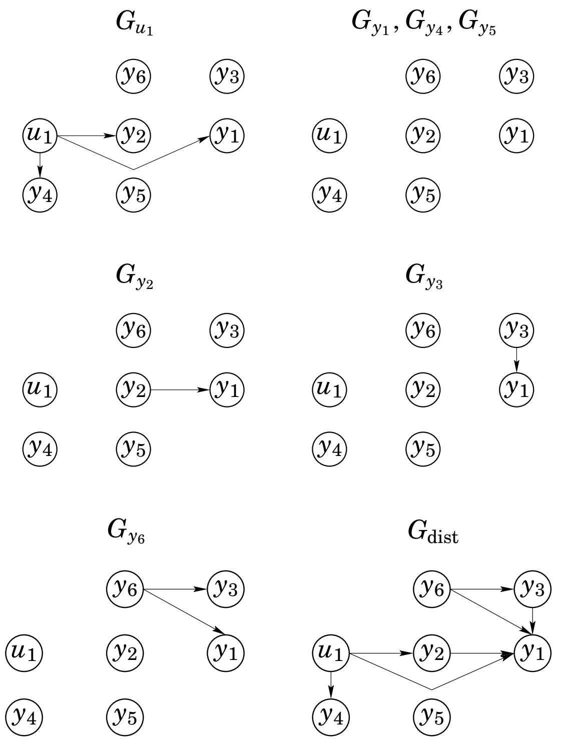

Fig. 3. Process graph identification for the water tank system.

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

Breaking up V like this is, of course, closely related to Kalman decomposition, see [5]. It is easy to compute V_{rr}, V_{ru}, V_{ur}, and V_{uu} from G (and the corresponding estimates from $\hat{G}$).

Example 1. (Cont'd). The process graph for the water tank system in Figure 2 has the reachability structure given by $V_{rr} = \{y_2\}$, $V_{ru} = \{y_4\}$, $V_{ur} = \{y_3, y_6\}$, and $V_{uu} = \{y_5\}$.

The process graph with $V = \{u_1, y_1, y_2\}$ and $E = \{(u_1, y_1), (y_2, y_1)\}$ is a feedforward prototype. Here $V_{ur} = \{y_2\}$ and $V_{rr} = V_{ru} = V_{uu} = \emptyset$. 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 may improve the attenuation of this particular disturbance in the control loop.

A process graph prototype for cascade control is given by $V = \{u_1, y_1, y_2\}$ and $E = \{(u_1, y_2), (y_2, y_1)\}$. 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. If there is only one vertex in the path between u_1 and y_1 that is used for feedback, then cascade control is the commonly used term for this control structure. When there are two or more measurements fed back to the controller, we simply say feedback control. This case corresponds to a control law based on (partial) state feedback.

By generalizing the conclusions from the previous three-vertex 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 related by fast dynamics; then it might be sufficient to use only one of them. Further practical considerations are discussed in [5].

Example 1. (Cont'd). Assume that the present control structure for the water tank problem is a SISO control loop consisting of u_1 and y_1. 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 final graph $\hat{G} = G$ in Figure 2 with $V_{rr} = \{y_2\}$, $V_{ru} = \{y_4\}$, $V_{ur} = \{y_3, y_6\}$, 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 for the water tank system. 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.

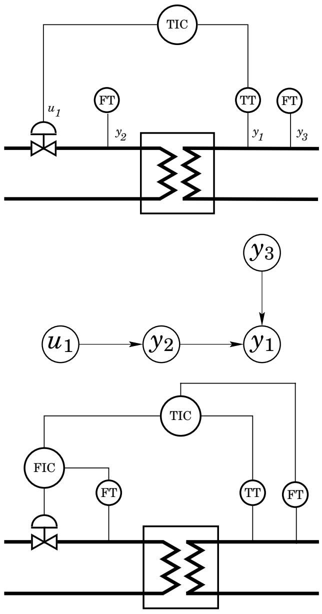

Fig. 4. Control of a heat exchanger. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

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 heat exchanger). A process diagram for the control of a heat exchanger is shown in the top diagram of Figure 4. 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 steps experiments in u_1, y_2, and y_3, the resulting process graph is as given by the middle graph in Figure 4. 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 diagram in the figure. This is the control configuration often used in practice for control of a heat exchanger.

Example 3. (Concentration control). A process diagram for a concentration control loop is shown in Figure 5. 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 5. This is, of course, the natural configuration for this control problem.

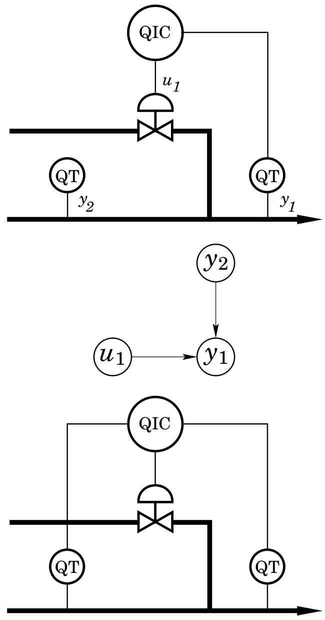

Fig. 5. Concentration control. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

Example 4. (Drum level control). Level control of a drum boiler is illustrated by the process diagram in Figure 6. The control objective is to control the level in the drum (y_1) using the feed-water 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 6. 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 (and possibly in y_1) 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 6 showing one of the standard configurations for drum level control.

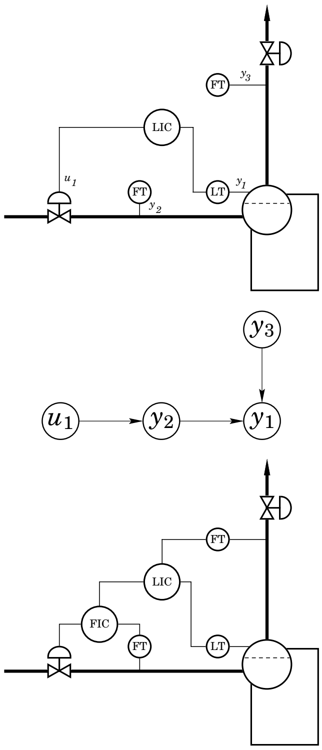

Fig. 6. Drum level control. Top diagram shows the original control loop, the middle shows the process graph, and the bottom diagram shows the modified control structure.

6. CONCLUSIONS

A control structure design algorithm was presented. No prior information about the plant dynamics was required, but a process graph model was obtained through a number of simple step experiments. The process graph illustrated the causal relations in the process and automatically suggested a control structure. The process graph might also be a pedagogical instrument to present control structures for process operators. In the generic algorithm presented here, experiments are done for all vertices in the graph except for the controlled process output. The algorithm can easily be modified in order to limit the number of performed experiments.

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.

Acknowledgment

The authors would like to thank Bo Bernhardsson for valuable comments. This work was supported by the Swedish Foundation for Strategic Research through its Center for Chemical Process Design and Control. Karl Henrik Johansson has also been supported by Swedish Foundation for International Cooperation in Research and Higher Education.

7. REFERENCES

[1] P. J. Antsaklis, K. M. Passino, and S. J. Wang. “An introduction to autonomous control systems.” *IEEE Control Systems Magazine*, **11:4**, pp. 5–13, June 1991.

[2] K. J. Åström. “Assessment of achievable performance of simple feedback loops.” *International Journal of Adaptive Control and Signal Processing*, **5**, pp. 3–19, 1991.

[3] D. G. Bobrow, Ed. *Qualitative Reasoning About Physical Systems*. MIT Press, Cambridge, MA, 1985.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[5] K. H. Johansson and T. Hägglund. “Control structure design in process control.” Report ISRN LUTFD2/TFRT--7585--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, August 1999.

[6] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1985.

[7] J. Montmain and S. Gentil. “Causal modeling for supervision.” In *Proceedings of IMACS*, 1999.

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

[9] F. G. Shinskey. *Controlling Multivariable Processes*. Instrument Society of America, Research Triangle Park, NC, 1981.

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

[11] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control—Analysis and Design*. John Wiley & Sons, 1996.