Session 12

Hierarchical Systems Theory

Jari Hätönen

Systems Engineering Laboratory, University of Oulu
P.O. BOX 4300, FIN-90014 University of Oulu, Finland

12.1 Introduction

The design and analysis of large and complex system requires frequently that the system is divided into smaller units, called sub-systems. The structure of overall system resulting from the interconnections of sub-systems can be very complex. One of the most common structures, is the *hierarchical* structure, i.e. the lay-out of the structure is vertical. Consequently in this report only hierarchical systems are considered, and a special emphasis is put on two-level hierarchical systems. A diagram of a standard two-level hierarchical system is shown in Fig. 12.1, where, as expected, two levels can be found, namely the lower level and the upper level. The lower level consists of the process level, where the process level has been divided into N sub-systems. The sub-systems are connected to each other because there is either material or information flows between these sub-systems. Each sub-system has its own decision unit, which tries to control the behaviour of the sub-system so that the objectives of this particular sub-system would be met. The decision unit can also use feedback information from the sub-system to improve its 'control policy'. However, quite often the objectives of the sub-systems are conflicting, resulting in a poor overall performance. Hence an upper-level decision unit

or a *coordinator* has to be introduced, and the objective of this decision unit is to coordinate the decision making of the sub-systems so that the overall performance of the system would be improved. The coordinator receives information from the sub-systems so that it can monitor the performance of the overall system. Note that this approach is not as restrictive as it sounds, because if a new higher level is added into a two-level system, in most of cases the new level can be chosen to be the higher level of the modified system, and the original two-level system becomes the lower-level of the modified system.

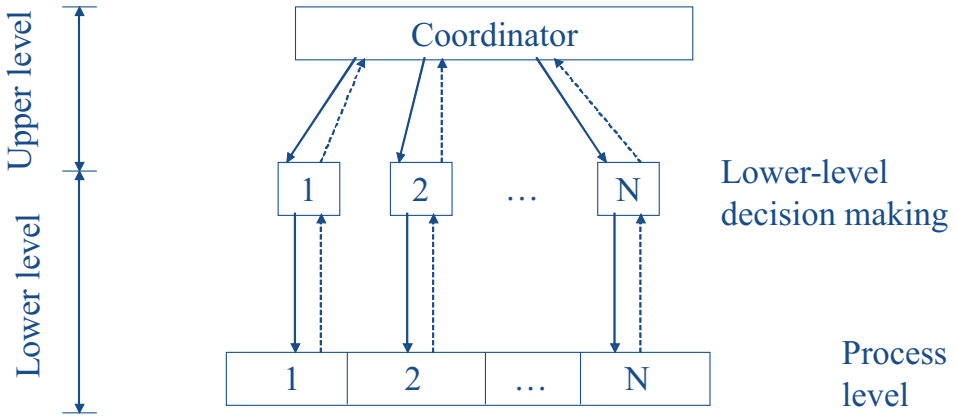

Figure 12.1: A diagram of a standard two-level hierarchical system

The strength of the two-level hierachical system theory is that there exists a large number of important systems around us that can be seen as being a two-level hierarchical system. Examples considered in [3] and [2] are organisation structures in large companies, distribution networks, oil refineries and power plants, to name a few. Consequently it is an interesting question why two-level hierarchical systems are so frequent. The following observations might be at least a partial answer to this question:

1) It is easier to analyse and (re)design large-scale systems if they are broken into smaller units.

2) The sub-system approach allows specialisation, where a sub-system is only responsible for its own task and does not require information of the objectives of the overall system.

3) Hierarchical systems allow a certain degree of fault tolerance. This is due to the fact that if a sub-system breaks down, the overall system does not necessarily stop working. Furthermore, due to 'module structure' the failure is 'localised' and hence easy to detect and and repair

(i.e. only the faulty module has to replaced and its connections re-established). The coordinator, however, is the weak point, because if it stops working, the overall system cannot function anymore.

3) Even evolution seems to favour two-level hierachical systems. For example in a human body the brain can be considered as being the coordinator, whereas the rest of the body forms the sub-system level.

4) In the evolution of organisations two-level hierachical systems play a major role. Even pre-historic tribes had a tribe leader, whom was responsible for coordinating the actions of individual tribe members in order to improve the overall well-being of the tribe.

In the following material it is shown how two-level hierarchical systems can be analysed mathematically. Furthermore, it is shown how optimisation techniques can be used to coordinate the running of a two-level hierarchical systems. The material is a rather mathematical, but in order to understand it, only a fairly modest background is needed in constrained optimisation theory. This report is based on [1], which is a nice introduction into the theory of hierarchical systems.

12.2 Process level

12.2.1 Sub-system models

The process level consists of N inter connected sub-systems. For each sub-system i there exists a set of admissible inputs I_i, the set of admissible outputs O_i and a sub-system model $f_i : I_i \rightarrow O_i$. Because the sub-systems are interconnected to each other, the set of inputs I_i is divided into the set of free inputs M_i and the set of interconnected inputs X_i and consequently

$$I_i \subset M_i \times X_i \tag{12.1}$$

In a similar fashion the the outputs are divided into the set of free ouput variables Y_i and set of constrained output variables Z_i, i.e. the outputs $z_i \in Z_i$ are fed as inputs into other sub-systems, see Fig. 12.2. With this division the set of admissible outputs O_i can be written as

$$O_i \subset Y_i \times Z_i \tag{12.2}$$

In order to achieve mathematical tractability, from now on it is assumed that $M_i, X_i, , Y_i, Z_i$ are suitable vector spaces. Typically the vector spaces would be chosen to be suitable vector spaces of time functions (this is of course highly application dependent), $L_2[0,T]$-spaces, $l_2[0,T]$ and $C_\infty[0,T]$ being frequently used spaces. In order to describe the interconnections present in the system an interconnection mapping $H : Z \to X$ is defined where

$$\begin{aligned} Z &:= Z_1 \times Z_2 \times \ldots \times Z_N \\ X &:= X_1 \times X_2 \times \ldots \times X_N \end{aligned} \tag{12.3}$$

and N is again the number of sub-systems.

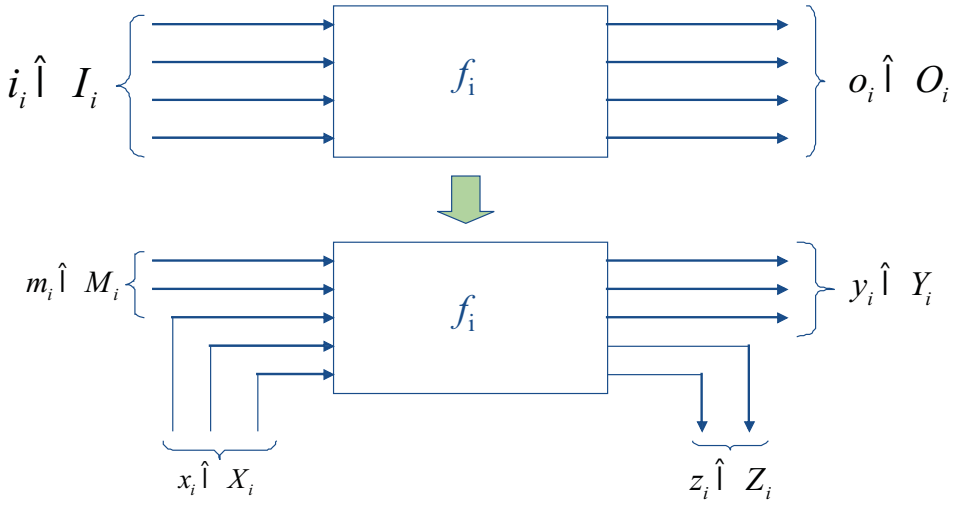

Figure 12.2: Division of variables into free and interconnected variables

Without loss of generality it can be shown that for each x_i, the corresponding interconnections can be written as

$$x_i = \sum_{i=1}^{N} C_{ij} z_j, \qquad i = 1, 2, \ldots, N \tag{12.4}$$

where $x_i \in \mathbf{R}^{n_i}$, $z_j \in \mathbf{R}^{m_j}$ and C_{ij} is a real-valued $n_i \times n_j$-matrix. Furthermore, the element c_{ij}^{kl} of C_{ij} is either zero or one depending on whether or not the lth component of z_j is connected to the kth element of input vector x_i. Define now

$$\begin{aligned} X &:= X_1 \times X_2 \times \ldots \times X_N \quad & M &:= M_1 \times M_2 \times \ldots \times M_N \\ Y &:= Y_1 \times Y_2 \times \ldots \times Y_N \quad & Z &:= Z_1 \times Z_2 \times \ldots \times Z_N \end{aligned} \tag{12.5}$$

and

$$\begin{array}{ll} x = (x_1, x_2, \ldots, x_N) & m = (m_1, m_2, \ldots, m_N) \\ y = (y_1, y_2, \ldots, y_N) & z = (z_1, z_2, \ldots, z_N) \end{array} \tag{12.6}$$

Define further the mapping $f : M \times X \supset D(f) \to Y \times Z$ and $C : Z \to X$ so that

$$f : (m, x) \to (y, z) = (y_1, \ldots, y_N, z_1, \ldots, z_N) \tag{12.7}$$

where

$$(y_i, z_i) = f_i(m_i, x_i), \qquad i = 1, 2, \ldots, N \tag{12.8}$$

and the interconnection mapping C where

$$C(z) = x = (x_1, x_2, \ldots, x_N) \tag{12.9}$$

and

$$x_i = \sum_{i=1}^{N} C_{ij} z_j \tag{12.10}$$

From now on it is *assumed* that exists a model $F : M \supset D(F) \to Y \times X \times Z$ so that for an arbitrary $m \in D(F)$ the equations

$$\begin{cases} (y, z) = f(m, x) \\ x = C(z) \end{cases} \tag{12.11}$$

define *uniquely* $x \in X$, $z \in Z$ and $y \in Y$ so that

$$F : m \to F(m) = (y, x, z) \tag{12.12}$$

Consequently F represents an overall process model shown in Fig. 12.3.

For future purposes the overall mapping F is divided into components $P : D(F) \to Y$, $K : D(F) \to X$, $S : D(F) \to Z$ so that

$$F(m) = (P(m), K(m), S(m)) \tag{12.13}$$

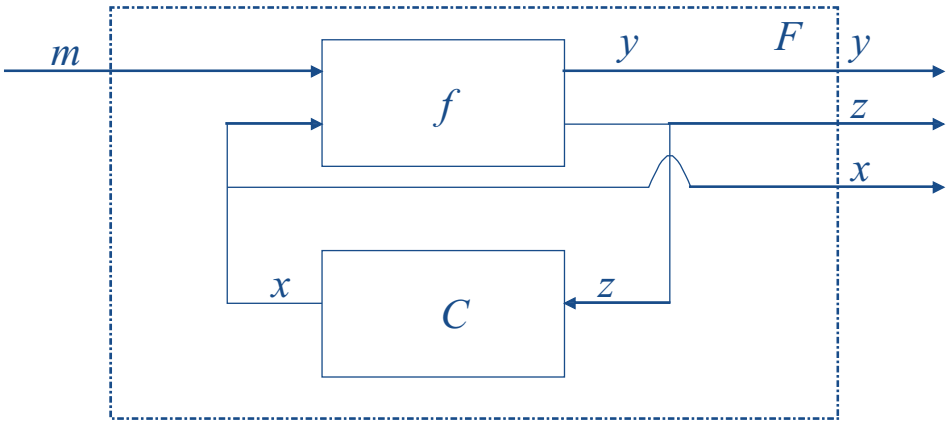

Figure 12.3: The overall mapping F

Remark 1 *Note that even the overall process model F exists, it is not necessarily explicitly available for calculations. This is due to the fact that in order to construct F the constrained input vector x has to be solved as a function of the free input vector m from the set of equations in (12.11). If the number of sub-systems is large and their mathematical models f_i are complex, it can be either impossible or impractical to solve F from (12.11).*

Remark 2 *In the context of dynamical systems the existence of F requires that the initial conditions of the system have to specified. Furthermore, the interconnection has to made so that the overall system model F is also causal.*

Note that P, K and S can written in the following component form

$$\begin{aligned} P &= (P_1, P_2, \ldots, P_N) \\ K &= (K_1, K_2, \ldots, K_N) \\ S &= (S_1, S_2, \ldots S_N,) \end{aligned} \tag{12.14}$$

where $P_i : D(F) \to Y_i$, $K_i : D(F) \to X_i, S_i : D(F) \to Z_i$, $i = 1, 2, \ldots, N$.

12.2.2 Lower-level decision units

As was explained earlier, for each sub-system i there exists a decision unit, and the objective of the decision unit is to control the sub-system according to its objectives by manipulating the free input variables m_i. The objectives of the lower-level decision units can vary a lot, but in this report each sub-system i is associated with a cost function g_i, which is a function of the

sub-systems input and output variables m_i, x_i, y_i and z_i. In other words for each sub-system i there exists a mapping (cost function)

$$g_i : M_i \times X_i \times Y_i \times Z_i \to \mathbf{R} \tag{12.15}$$

Furthermore, the sub-system model f_i can be used to eliminate the output variables from the cost function, and the cost function becomes only a function of m_i and x_i in the following way:

$$G_i : M_i \times X_i \supset D(G_i) \to \mathbf{R}, \qquad D(G_i) = D(f_i) \tag{12.16}$$

where

$$G_i(m_i, x_i) = g_i\left(m_i, x_i, f_i^1(m_i, x_i), f_i^2(m_i, x_i)\right) \tag{12.17}$$

with the notation $f_i(m_i, x_i) = (f_i^1(m_i, x_i), f_i^2(m_i, x_i))$. In a similar fashion an overall cost function g is defined where

$$g : M \times X \times Y \times Z \to \mathbf{R} \tag{12.18}$$

The overall process model can then used to eliminate the output variables resulting in equivalent representation of (12.18)

$$\begin{aligned} &G : M \supset D(G) \to \mathbf{R}, \qquad D(G) = D(F) \\ &G(m) = g\left(m, K(m), P(m), S(m)\right) \end{aligned} \tag{12.19}$$

From now on it is *assumed* that the individual cost functions G_i and the overall cost funtion G are related to each other with the equation

$$G(m) = \sum_{i=1}^{N} G_i\left(m_i, K_i(m)\right) \tag{12.20}$$

To judge whether or not this is always plausible is left to the reader.

12.3 Upper level

12.3.1 Upper-level decision making

The objective of the coordinator is affect the lower-level decision making so that that the cost function $G : D(F) \to \mathbf{R}$ is minimised, i.e. that the overall

optimisation problem is being solved. The overall optimisation problem can be written equivalently as a constrained optimisation problem

$$\min_{m \in D(F)} \sum_{i=1}^{N} G_i(m_i, x_i) \tag{12.21}$$

with the constraint

$$x = K(m) \tag{12.22}$$

This formulation, unfortunately, requires that the overall process model F is explicitly available, and as was earlier explained, this is not the case if the system contains a large number of complex sub-systems.

Consequently the overall optimisation problem has to modified so that it can be divided into independent optimisation problems for each sub-system. In order to achieve this, define a modified sub-system model $\tilde{f}_i$ and cost function $\tilde{g}_i$, $i = 1, 2, \ldots, N$ that now become a function of an external parameter $\gamma \in \Gamma$, where γ is the *coordination* parameter:

$$\begin{array}{ll} \tilde{f}_i : M_i \times X_i \times \Gamma \supset D(\tilde{f}_i) \rightarrow Y_i \times Z_i, & D(\tilde{f}_i = D(f_i) \times \Gamma \\ \tilde{g}_i : M_i \times X_i \times Y_i \times Z_i \times \Gamma \rightarrow \mathbf{R} & \end{array} \tag{12.23}$$

Furthermore, the modified sub-system model can be used to eliminate the output variables from the modified cost function, resulting in

$$\tilde{G}_i(m_i, x_i, \gamma) = \tilde{g}_i \left(m_i, x_i, \tilde{f}_i^1(m_i, x_i, \gamma), \tilde{f}_i^2(m_i, x_i, \gamma), \gamma \right) \tag{12.24}$$

This results in the following sub-system decision making problem

12.3.2 Sub-system decision making problem

The sub-system decision unit i has optimise with a *given* $\gamma \in \Gamma$ the cost function $\tilde{G}_i(m_i, x_i, \gamma)$, i.e. the decision unit has to find $(m_i, x_i) \in M_i \times X_i$ so that resulting pair (m_i, x_i) minimises the cost function $\tilde{G}_i(m_i, x_i, \gamma)$.

Remark 3 *The important point here is that now the optimisation problem is now an unconstrained optimisation problem, and the fact that x_i is determined by the behaviour of other sub-systems is not taken into account.*

The optimal solution of the problem (if it exists in the first place) is called as the γ-optimal solution $(m(\gamma), x(\gamma))$, where

$$m(\gamma) = (m_1(\gamma), m_2(\gamma), \ldots, m_N(\gamma)) \tag{12.25}$$

and

$$\begin{array}{l} \min_{m_i, x_i} \tilde{G}_i(m_i, x_i, \gamma) = \tilde{G}_i(m_i(\gamma), x_i(\gamma), \gamma) \\ i = 1, 2, \ldots, N \end{array} \tag{12.26}$$

The objective of the coordinator is select the coordination variable $\gamma \in \Gamma$ so that the overall optimisation is being minimised, i.e.

$$\min_{m \in D(F)} G(m) = G\left(m(\gamma)\right) \tag{12.27}$$

The question whether or not there exits a γ so that (12.27) holds depends strongly on how the modified 'system variables' $\tilde{f}_i$, $\tilde{G}_i$ and the set of coordination variables Γ are chosen. Consider now the case where the modification is done so that for an arbitrary $\gamma \in \Gamma$, $m \in D(F)$ it holds that

$$\begin{array}{l} \tilde{f}_i(m_i, K_i(m), \gamma) = f_i\left(m_i, K_i(m)\right) \\ \tilde{G}_i(m_i, K_i(m), \gamma) = G_i\left(m_i, K_i(m)\right) \end{array} \tag{12.28}$$

i.e. the modified sub-system model is equivalent to the original sub-system model and the modified cost function is equivalent to the original cost function when the constraint equation $x = K(m)$ is met.

Proposition 1 *Suppose that sub-system model f_i and the sub-system cost function G_i are modified according to (12.28) and htat there exists $m^\circ \in D(F)$ and $\gamma^\circ \in \Gamma$ so that*

$$\begin{array}{l} \min_{m \in D(F)} G(m) = G(m^\circ) \\ x(\gamma^\circ) = K\left(m(\gamma^\circ)\right) \end{array} \tag{12.29}$$

This implies that $G(m^\circ) = G\left(m(\gamma^\circ)\right)$.

Proof.

$$\begin{array}{l} G\left(m(\gamma^\circ)\right) = \sum_{i=1}^N G_i(m_i, K_i\left(m(\gamma^\circ)\right) \\ = \sum_{i=1}^N \tilde{G}_i(m_i(\gamma^\circ), K_i(m_i(\gamma^\circ), \gamma^\circ) \\ = \sum_{i=1}^N \min_{m_i, x_i \in D(f_i)} \tilde{G}_i(m_i, x_i, \gamma^\circ) \\ \leq \sum_{i=1}^N \min_{m \in D(F)} \tilde{G}_i(m_i, K_i(m), \gamma^\circ) \\ \leq \min_{m \in D(F)} \sum_{i=1}^N \tilde{G}_i(m_i, K_i(m), \gamma^\circ) \\ \leq \sum_{i=1}^N \tilde{G}_i(m_i^\circ, K_i(m^\circ), \gamma^\circ) \\ = \sum_i^N G_i(m_i^\circ, K_i(m^\circ) = G(m^\circ) \end{array} \tag{12.30}$$

and hence $G\left(m(\gamma^\circ)\right) \leq G(m^\circ)$. However, by optimality, $G\left(m(\gamma^\circ)\right) \geq G(m^\circ)$, and consequently $G\left(m(\gamma^\circ)\right) = G(m^\circ)$, which concludes the proof.

Note that even if the modification is done as in (12.28), is does not guarantee that condition (12.27) is met because in the proof it is *assumed* that there exists at least one $\gamma^\circ \in \Gamma$ so that resulting pair $(m(\gamma^\circ), x(\gamma^\circ)))$ satisfies the 'interconnection equation' $x(\gamma^\circ) = K\left(m(\gamma^\circ)\right)$. Whether or not this is true leads to the definition of coordinability, which is the topic of the next section.

12.4 Coordination

As was explained in the previous section the sub-system decision problems were defined to be optimisation problems where the optimisation problems can be modified with an external parameter $\gamma \in \Gamma$. The purpose of this modification is to make the sub-system decision problems independent from each other and to remove the 'conflicts' caused by the interconnections between the sub-systems. Furthermore, the objective of the upper-level coordinator is to find a $\gamma^\circ \in \Gamma$ so that with this particular γ° the solutions of the sub-system optimisation problems also satisfy the interconnection equations and solve the overall optimisation problem. This results in the following definition of coordinability:

Definition 1 (Coordinability) *1) The overall optimisation problem has a solution, i.e.* $\exists m^\circ \in D(F)$ *so that*

$$\min_{m \in D(F)} G(m) = G(m^\circ) \tag{12.31}$$

ii) There exists $\gamma^\circ \in \Gamma$ *so that there exists a* γ°*-optimal pair* $(m(\gamma^\circ), x(\gamma^\circ))$*, i.e.*

$$\begin{array}{l} \min_{(m_i, x_i) \in D(f_i)} \tilde{G}_i(m_i, x_i, \gamma^\circ) = \tilde{G}_i(m_i(\gamma^\circ), x_i(\gamma^\circ), \gamma^\circ) \\ i = 1, 2, \ldots, N \end{array} \tag{12.32}$$

iii) The objectives of the sub-system decision units and upper-level decision unit are in 'harmony', i.e. $m(\gamma^\circ) \in D(F)$ *and*

$$G(m^\circ) = G\left(m(\gamma^\circ)\right) \tag{12.33}$$

As a consequence the coordinability of the overall system guarantees that there exists at least one $\gamma \in \Gamma$ so that the γ-optimal of the sub-system optimisation problem is equal to solution of the overall optimisation problem. The coordinability of two-level hierachical system, however, does not lead into an efficient solution method, because the coordinator has to know a'priori the optimal γ° that results in the solution of the overall optimisation problem.

If the coordinator cannot choose directly the optimal value γ°, which is the case in most problems, the coordination becomes an *iterative* process where the coordination variable is changed iteration by iteration to the 'correct' direction. In order to implement the 'correction process' the coordinator needs a *coordination strategy*. In the coordination strategy the new value for the coordination variable depends on the current value of the coordination variable and the corresponding γ-optimal solution $(m(\gamma), x(\gamma))$, resulting in the coordination strategy η,

$$\eta : \gamma \times M \times X \to \Gamma \tag{12.34}$$

The coordination algorithm related to the coordination strategy can be described in the following way:

1) Select a suitable initial guess for $\gamma \in \Gamma$

2) The lower-level decision units solve their own optimisation problems resulting in the γ-optimal pair $(m(\gamma), x(\gamma))$

3) If γ is not optimal (whether or not this can be tested in practise depends highly on the modification technique, see next section for further details), the coordinator selects a new coordination variable $\gamma \in \Gamma$ using the coordination strategy

$$\gamma \leftarrow \eta\left(\gamma, m(\gamma), x(\gamma)\right) \tag{12.35}$$

and the algorithm jumps back to 2).

Note this algorithm is purely abstract and due to its abstractness it is impossible to analyse whether not the coordination algorithm converges. In this next section one possible way to implement the coordination strategy is being discussed.

12.5 The balancing principle

The balancing principle is coordination method where the interconnections between sub-systems are 'cut off' resulting in N independent optimisation problem. Each sub-system decision unit i optimises its running not only as a function of m_i but also a function of the interconnected input variable x_i. Because now the sub-systems optimise their running independently from each other, the constraint equation $x_i = \sum_{i=1}^{N} C_{ij} z_j$, $i = 1, 2, \ldots, N$ is not typically met, and the overall optimisation problem remains unsolved. Consequently in the balancing principle the sub-system decision units are forced to select optimal solutions (m_i, x_i) so that constraint equation is met.

The modification of the 'system variables' in the balancing technique is done in the following way:

$$\begin{aligned} &\tilde{f}_i(m_i, x_i, \gamma) := f_i(m_i, x_i) \\ &\tilde{g}_i(m_i, x_i, y_i, z_i, \gamma) := g_i(m_i, x_i, y_i, z_i) + \psi_i(x_i, z_i, \gamma) \\ &\tilde{G}_i(m_i, x_i, \gamma) := \tilde{g}_i(m_i, x_i, f_1^1(m_i, x_i), f_i^2(m_i, x_i), \gamma) \end{aligned} \tag{12.36}$$

where the mappings $\psi_i : X_i \times Z_i \times \Gamma \to \mathbf{R}$ for $i = 1, 2, \ldots, N$ are defined so that

$$\psi(x, z, \gamma) := \sum_{i}^{N} \psi_i(x_i, z_i, \gamma) = 0 \tag{12.37}$$

if the balance equation

$$x_i = \sum_{i=1}^{N} C_{ij} z_j, \qquad i = 1, 2, \ldots, N \tag{12.38}$$

is met. Hence in the balance principle only the cost functions are modified whereas the sub-system model is equal to the original sub-system model. Note that modification that satisfies (12.37) and (12.38) is called a *zero-sum modification*. The defining property of a zero-sum modification is that the effect of the modification disappears from the overall cost function $\sum_{i=1}^{N} G_i$ when the system is in 'balance', i.e. the interconnection equations hold. In this case the overall cost can be calculated as the sum of the individual values of the cost functions in the following way:

$$G(m) = \sum_{i=1}^{N} \tilde{G}_i(m_i, x_i, \gamma) = \sum_{i=1}^{N} G_i(m_i, x_i) \tag{12.39}$$

How to select the coordination variable $\gamma \in \Gamma$ is not discussed here. However, in the next section it is shown how Langrange multiplier theory can be used to implement the balancing principle. In the balancing principle the decision problem for each sub-system decision unit is given by

$$\min_{(m_i,x_i)} \tilde{G}_i(m_i, x_i, \gamma) = \tilde{G}_i(m_i(\gamma), x_i(\gamma), \gamma) \tag{12.40}$$

Define now a mapping $\phi_i : \Gamma \supset D(\phi_i) \to \mathbf{R}$,

$$\phi_i(\gamma) = \min_{(m_i,x_i)\in D(f_i)} \tilde{G}_i(m_i, x_i, \gamma) \tag{12.41}$$

where

$$D(\phi_i) = \{\gamma \in \Gamma | \phi_i(\gamma) \text{ exists}\} \tag{12.42}$$

Consider now the modified cost function that can be written as

$$\begin{array}{l} \sum_{i=1}^{N} \tilde{G}_i(m_i, x_i, \gamma) \\ = \sum_{i=1}^{N} g_i(m_i, x_i, f_i^1(m_i, x_i), f_i^2(m_i, x_i) + \sum_{i=1}^{N} \psi_i(x_i, f_i^2(m_i, x_i), \gamma) \end{array} \tag{12.43}$$

If the original overall optimisation problem has a solution m° and there exists a $\gamma^\circ \in \cap D(\phi_i)$ so that the lower-level γ°-optimal solution $(m(\gamma^\circ), x(\gamma^\circ))$ satisfies

$$x(\gamma^\circ) = \sum_{i=1}^{N} C_{ij} f_j^2 \left(m_j(\gamma^\circ), x_j(\gamma^\circ)\right), \qquad i = 1, 2, \ldots, N \tag{12.44}$$

or in a more explicit form

$$x(\gamma^\circ) = K(m\,(\gamma^\circ)) \tag{12.45}$$

then it holds that

$$\begin{array}{l} \phi(\gamma^\circ) = \sum_{i=1}^{N} \phi_i(\gamma^\circ) = \sum_{i=1}^{N} \tilde{G}_i\left(m_i(\gamma^\circ), x_i(\gamma^\circ), \gamma^\circ\right) \\ = \sum_{i=1} G_i\left(m_i(\gamma^\circ), x_i(\gamma^\circ)\right) = G\left(m(\gamma^\circ)\right) \end{array} \tag{12.46}$$

because of the zero-sum modification. On the other hand $\forall \gamma \in D(\phi)$

$$\begin{array}{l} \phi(\gamma) = \min_{(m,x)\in D(f)} \sum_{i=1}^{N} \tilde{G}_i(m_i, x_i, \gamma) \\ \leq \min_{(m,x)\in D(f), x=K(m)} \sum_{i=1}^{N} \tilde{G}_i(m_i, x_i, \gamma) \\ = \min_{(m,x)\in D(f), x=K(m)} \sum_{i=1}^{N} G_i(m_i, x_i) = G(m^\circ) \\ \leq G\left(m(\gamma^\circ)\right) \end{array} \tag{12.47}$$

and was shown previously in Section 12.3, $G(m^\circ) = G\,(m(\gamma^\circ))$. Consequently in tha balancing principle the coordinator variable γ is taken to be the solution of the maximisation problem

$$\max_{\gamma \in D(\phi)} \phi(\gamma) \tag{12.48}$$

In practise it can be difficult (impractical) to solve analytically this maximisation problem, and numerical methods have to be used instead. Typically the gradient $\nabla\phi$ is available (see next section), and for example a steepest-descent algorithm can be used to solve iteratively the maximisation problem.

12.6 The Langrange technique

In the Langrange technique the overall optimisation problem is modified by adjoining the interconnection equation into the cost function, resulting in the following Langrange function

$$L(m,x,y) = \sum_{i=1}^{N} G_i(m_i,x_i) + \sum_{j=1}^{N} \left\langle \gamma_i, x_i - \sum_{i=1}^{N} C_{ij} f_j^2(m_j,x_j) \right\rangle \tag{12.49}$$

where $< \cdot,\cdot >$ is the inner product in X_i (for simplicity it is assumed here that X_i for $i = 1,2,\ldots,N$ is always the same space and that it is reflexive, i.e. the dual space of X_i is X_i. Reflexive spaces are for example the Euclidian space $\mathbf{R}^N$ (discrete-time case with finite time-axis) and $L_2[0,T]$ (continuous-time case with finite time-axis), which are one of the most commonly used spaces in control theory. By changing the summation order it can be shown that the Langrange function can be written equivalently as

$$\begin{array}{l} L(m,x,\gamma) = \sum_{i=1}^{N} L_i(m,x,\gamma) \\ L_i(m,x,\gamma) = G_i(m,x_i) + \langle \gamma_i, x_i \rangle - \sum_{j=1}^{N} \langle \gamma_j, C_{ij} f_i^2(m_i,x_i) \rangle \end{array} \tag{12.50}$$

In summary $L(m,x,\gamma)$ has been divided into a sum where each term $L_i(m_i,x_i,\gamma)$ depends only on the variables related to sub-system i and the Langrange multiplier γ. This modification is clearly a zero-sum modification. As was shown in the previous section, in the balancing technique the sub-system decision problem is to solve with a given γ (which is now the Langrange-multpier) the optimisation problem

$$\phi_i(\gamma) := \min_{m_i,x_i \in D(f_i)} L_i(m_i,x_i,\gamma) = L_i(m_i(\gamma),x_i(\gamma),\gamma) \tag{12.51}$$

and upper-level decision problem is

$$\max_{\gamma \in D(\phi)} \phi(\gamma) \tag{12.52}$$

where (as previously)

$$\phi = \sum_{i=1}^{N} \phi_i \tag{12.53}$$

As was mentioned previously, the upper level decision problem does not necessarily have a nice closed-form solution. Consequently in order to use numerical optimisation methods, the gradient $\nabla \phi_i$ is needed. However, in the Langrange technique this is just (can you show this?)

$$\nabla_\gamma \phi(\gamma_o) = [\epsilon_1, \epsilon_1, \ldots, \epsilon_N]^T := \epsilon \tag{12.54}$$

where

$$\epsilon_i = x_i(\gamma_o) - C_{ij} f_j^2 \left(m(\gamma_o), x_j(\gamma_o)\right) \tag{12.55}$$

and consequently the coordination strategy can be chosen to be

$$\eta(\gamma, m, x) = \gamma + k \cdot \epsilon \tag{12.56}$$

where k is a step-length parameter the algorithm designer has to select. This results in the following algorithm:

1) Set $k = 0$ and select an initial guess $\gamma_k = \gamma_0$.

2) Solve the sub-system optimisation problems with $\gamma = \gamma_k$.

3) If the system is in balance, stop, the overall optimisation problem has been solved. Otherwise set $k \to k + 1$ and update

$$\gamma_{k+1} = \gamma_k + k \cdot \nabla_\gamma \phi(\gamma_k) \tag{12.57}$$

and go to step 2.

Remark 4 *If sub-system decision problems are linear programmes (i.e. they are of the form $c^T x$ then it can be shown that the Langrange multipier can be understood to be the price that a sub-system has to pay to the other sub-system for the transfer of material (information). In other words coordination is equivalent to 'an optimal pricing policy between departments', see [3]!*

The toy example in the following section will illustrate the Langrange approach. Note that even this example considers only static sub-system models, this technique can be applied on dynamical systems models without any major complications, see [2] and [1] for further details.

12.7 A toy example

Consider the linear static system

$$\begin{cases} y_1 = 2m_1 + u_1 = P_1(m_1, u_1) \\ y_2 = 2m_2 + u_2 = P_2(m_2, u_2) \end{cases} \tag{12.58}$$

with the interconnections $u_1 = m_2$, $u_2 = m_1$, and the overall cost function is defined to be

$$G(m, y) = m_1^2 + m_2^2 + (y_1 - 1)^2 + (y_2 - 2)^2 \tag{12.59}$$

It is a straightforward exercise to show that the optimal solution for the optimisation problem is $\hat{m} = [1/5\ 7/10]^T$. The corresponding Langrange function is given by

$$L(m, y, \gamma) = m_1^2 + m_2^2 + (y_1 - 1)^2 + (y_2 - 2)^2 + \gamma_1(u_1 - m_2) + \gamma_2(u_2 - m_1) \tag{12.60}$$

resulting in following two sub-system cost functions

$$\begin{array}{l} G_1(m_1, y_1, \gamma) = m_1^2 + (y_1 - 1)^2 + \gamma_1 u_1 - \gamma_2 m_1 \\ G_1(m_2, y_2, \gamma) = m_2^2 + (y_2 - 2)^2 + \gamma_2 u_2 - \gamma_1 m_2 \end{array} \tag{12.61}$$

For a fixed γ the optimal control policies for P_1 and P_2 become

$$\begin{bmatrix} 10 & 4 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ u_1 \end{bmatrix} = \begin{bmatrix} 4 + \gamma_2 \\ 2 - \gamma_1 \end{bmatrix}, \begin{bmatrix} 10 & 4 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} m_2 \\ u_2 \end{bmatrix} = \begin{bmatrix} 8 + \gamma_1 \\ 4 - \gamma_2 \end{bmatrix} \tag{12.62}$$

The gradient of $\phi(\gamma)$ becomes

$$\nabla_\gamma \phi(\gamma) = \begin{bmatrix} u_1 - m_2 \\ u_2 - m_1 \end{bmatrix} \tag{12.63}$$

and the update law for the coordination variable (Langrange multplier) becomes

$$\begin{bmatrix} \gamma_1(k+1) \\ \gamma_2(k+1) \end{bmatrix} = \begin{bmatrix} \gamma_1(k) \\ \gamma_2(k) \end{bmatrix} + k \begin{bmatrix} u_1 - m_2 \\ u_2 - m_1 \end{bmatrix} \tag{12.64}$$

where $k = 0.1$ (a sophisticated guess). The initial guess for γ is $\gamma(0) = [10\ 10]^T$. Fig. 12.4 shows how the 'input functions' converge, i.e. in this figure the Euclidian norm of $[m_1 - 1/5\ m_2 - 7/10]^T$ is plotted as function of the iteration rounds, showing a reasonable convergence speed. Fig. 12.5 on the other hand shows the value of the modified cost function. From this figure it can be seen that the coordinator is maximising the modified cost function, which is the expected result.

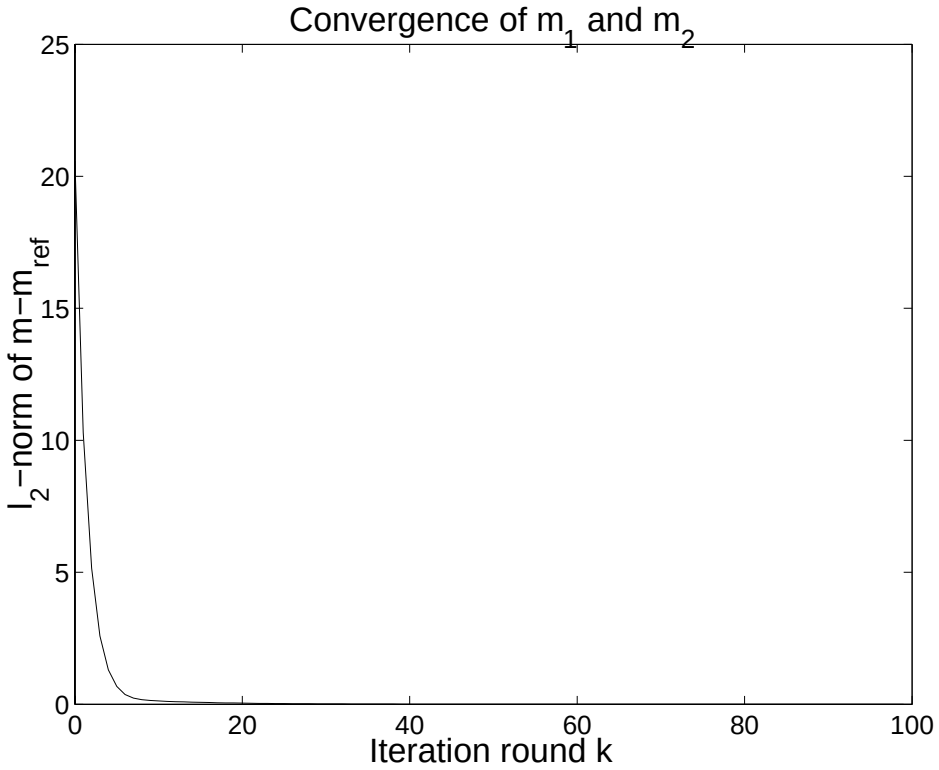

Figure 12.4: Convergence of the inputs

12.8 Conclusions

In this chapter a general theory for the optimisation of two-level hierarchical systems has been introduced. This theory can be applied on a wide range of applications, examples being economics, organisation theory and large-scale industrial plants. In this theory the system is divided into sub-systems, where each sub-system tries to achieve its own objectives without considering whether or not they contradict with other sub-systems. In order to rectify

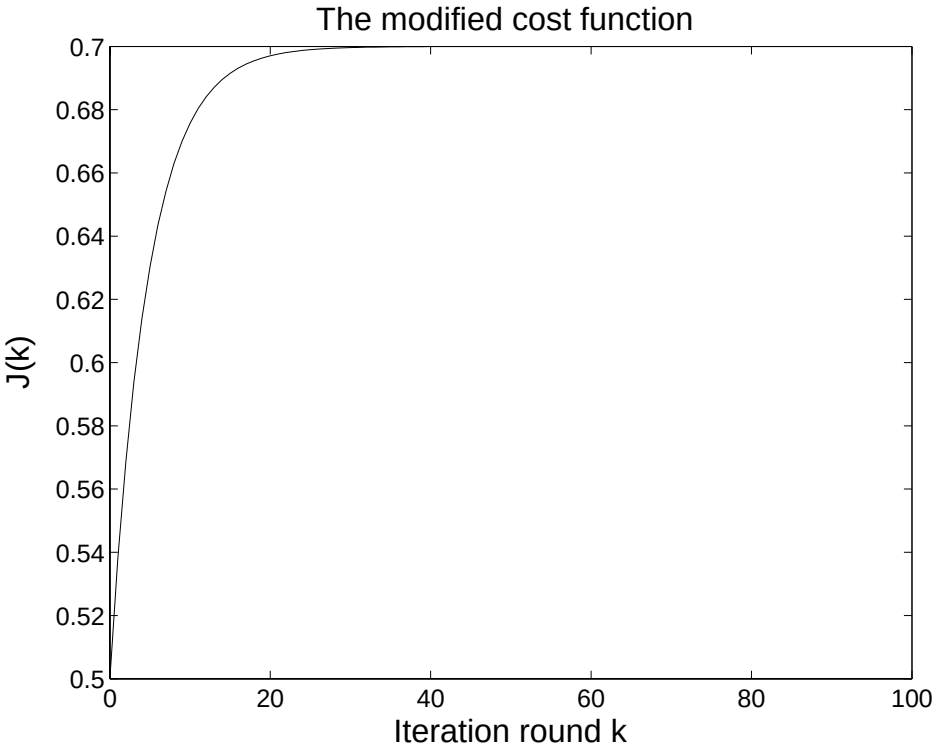

Figure 12.5: The value of the modified cost function

the 'selfishness' of the sub-system decision making, an upper-level decision making system has to be introduced. The objective of the upper-level decision making unit is to coordinate the decision making of each sub-system, so that an overall harmony is achieved. A special emphasis was put on the so called balancing technique, because it can be implemented numerically with the well-known Langrange technique. Note that in this report no rigorous convergence theory was presented for the iterative balancing technique. Convergence, however, can be proved in some special cases, see [2] for further details.

The required theoretical background for understanding the theory of two-level hierarchical systems in the most abstract setting is, unfortunately, reasonably sophisticated. Consequently the theory can be quite hard for an engineer to digest, and therefore it has not found its way to the mainstream material taught at graduate level in control engineering courses. On the other hand it offers an interesting option for more mathematically oriented engineers to do high-quality control for complex systems without resorting to ad-hoc methods (i.e. fuzzy control, agents etc.).

Bibliography

[1] L. Hakkala, Hierarkisten järjestelmien optimointi: koordinointiperiaatet, Report C 23, Systems Theory Laboratory, Helsinki University of Technology, 1976

[2] Mesarovic, Macko, Takahara, Theory of Hierarchical Multilevel Systems, Academic Press, New York, 1970

[3] Lasdon, Optimisation Theory for Large Systems, The Macmillan Publishing Company, New York, 1970