

Chapter 6

Towards Advanced Applications



6.1 Modelling Users and Their Interactions

Unlike automated assembly systems, human operated manufacturing and assembly ones encounter various unappealing effects associated with human behaviour uncertainty. The objective of this section is to provide two examples pertaining to modeling human users and their interactions [1] with the fuzzy logic approach [2–4]. The fuzzy logic paradigm seems to be a natural modelling tool for such a task as its origins stem from human inference behaviour. Indeed, fuzzy logic has been used for modelling human reliability in the process industry [5–7] as well as human error analysis [8]. There are also works utilising fuzzy logic for human factor modelling in preventive maintenance actions [9] and estimating a context-specific human error rate [10].

6.1.1 Assembly Process

Let us proceed to the first case, i.e., modelling human behaviour in the assembly process with fuzzy logic. For that purpose a pair of KIS.Devices is employed, i.e., KIS.BOX and KIS.LIGHT. For the purpose of illustration, let us consider a sample manual assembly process, which involves two phases [11] (see Fig. 6.1):

1. battery cell mounting,
2. cell-controller linking.

A general overview of the manual assembly station with the KIS.ME infrastructure is presented in Fig. 6.2. Let us recall that KIS.LIGHT possesses two digital inputs, which can be connected with two photoelectric sensors (see Sect. 3.1 for an illustrative example). Let us imagine that there are transportation means, e.g., conveyor belts, which provide battery cells and cell controllers. Table 6.1 presents all possible situations related to this. Using KIS.MANAGER Rule engine (cf. Sect. 2.10), one can implement rules governing such behaviour of the KIS.LIGHT operational LED

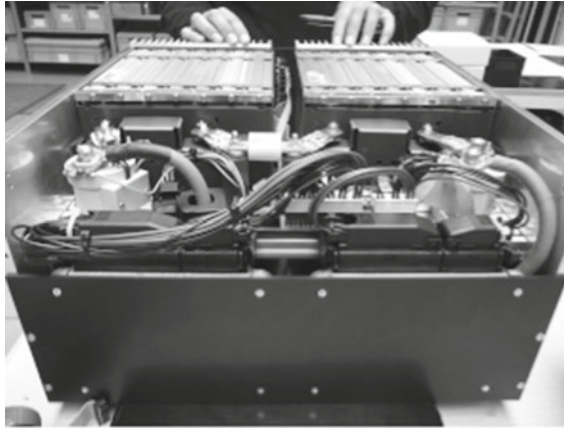


Fig. 6.1 Manual battery assembly

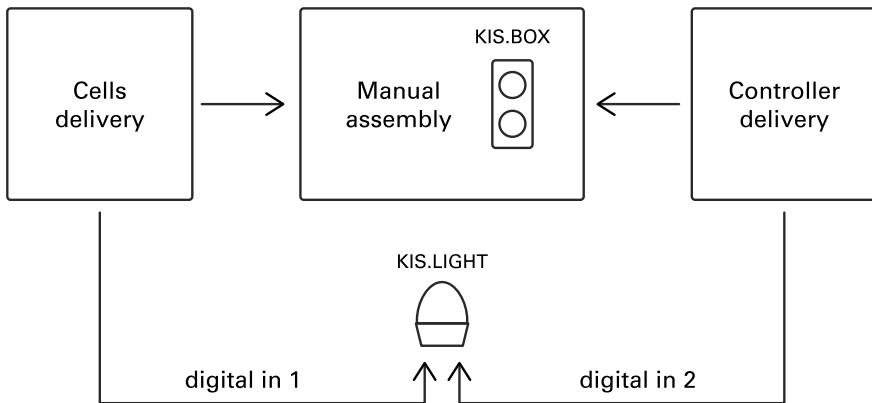


Fig. 6.2 General overview of the manual assembly station with the KIS.ME infrastructure

depending on its digital inputs. As a result, if its color is red, then there are no components. If one of them is available, then the color is green. Finally, if both of them are available, then the color is blue and it alarms the worker that the assembly process can be started. Thus, a necessary condition for mounting the battery is that all components be available. Let us proceed to describe all possible working states, which are listed in Table 6.2. Similarly as in the KIS.LIGHT case, the transition between States 1–5 can be realized using Rule engine. Moreover, as a transition trigger, one can use a KIS.BOX Button 1 pressing action. This action ends the deployment of KIS.Devices to the above assembly station. Note that the structure presented in Fig. 6.2 is very general, and hence it can be tailored to a wide range of assembly tasks. Having the KIS.Device infrastructure, it is possible to monitor the performance of the human operated assembly system using visualization, SPC and OEE strategies, described in

Table 6.1 KIS.LIGHT states

KL operational LED color	KL digital input 1	KL digital input 2
Red	0	0
Green	1	0
Green	0	1
Blue	1	1

Table 6.2 KIS.BOX states

State	KB Button 1 operational LED	KB Button 2 operational LED	Action
1	Blue	Blue	Can start
2	Red	Blue	Cell mounting
3	Green	Blue	Cell mounting completed
4	Green	Red	Cell-controller linking
5	Green	Green	Mounting completed

Chaps. 4–5. This is, however, beyond the scope of this section. Indeed, the problem is to obtain a time-driven model of assembly operations, which will include information about human experience and performance related with the working time within the shift.

For that purpose, let us recall that the time evolution of all states can be easily measured within KIS.MANAGER using the Data trend chart widget (cf. Sect. 2.7), which allows real-time analysis of operator performance. The historical data can easily be saved to a CSV file, and hence it can be further processed in external software. Thus, the objective is to model human behaviour taking into account the following issues:

- varying human experience and performance,
- a variety of batteries to be mounted.

Thus, the crucial problem is to model the realization times of States 2 and 4 in Table 6.2. For that purpose, two general models are introduced:

$$c_m = f_m(n_c, C_c, m_p, e_x, t_s), \tag{6.1}$$

$$c_l = f_l(n_c, e_x, t_s), \tag{6.2}$$

where

- c_m is the cell mounting time;
- c_l signifies the controller linking time;
- $f(\cdot)$ is the model structure;
- e_x represents the experience of an operator, e.g., in the range between 0 and 10;

Table 6.3 Fuzzy premise variables

Variable	Description	Intervals
e_x	Advanced	6–10
	Intermediate	3–7
	Beginner	0–4
t_s	Long	5–8
	Medium	3–6
	Short	0–4

- t_s is the working time within the shift, e.g., from 0 to 8 h;
- n_c stands for the number of cells to be mounted inside the battery pack;
- m_p is the mass of the battery pack.

Having all necessary ingredients, it is possible to provide the structure of (6.1)–(6.2). For that purpose, it is proposed to use the Takagi–Sugeno (TS) fuzzy logic approach [12, 13]. TS models are widely used for various modelling tasks (see [13] and the references therein). Their attractiveness is especially important in the case considered since there are two linguistic variables, presented in Table 6.3. They can be directly obtained, and hence they can form the so-called premise variables [13] for the TS counterparts of (6.1)–(6.2). According to the above KIS.Device infrastructure, each worker is associated with KIS.BOX, and hence its experience e_x can be directly obtained. As can be observed in Table 6.3, fuzzy logic allows smooth transitions between the groups. Indeed, with a degree of membership [13], a worker can be in one group. Similarly, the worker can be in a different group with another membership degree. The same feature pertains to the working time within the shift.

Under these preliminaries, the TS model of (6.1) is as follows:

$$\text{IF } t_s \in \mathbb{T}_{s,j} \text{ and } e_x \in \mathbb{E}_{x,i}, \text{ THEN}$$

$$c_m = p_{1,m}^{i,j} n_c + p_{2,m}^{i,j} m_p + p_{3,m}^{i,j} C_c, \quad i, j = 1, \dots, 3, \quad (6.3)$$

where p_i are the model parameters which have to be determined while $\mathbb{E}_{x,j}$ and $\mathbb{T}_{s,i}$ are fuzzy sets associated with Table 6.3. Each set can be shaped with three membership functions, which can be, e.g., of the Gaussian form:

$$\mathcal{G}^i(e_x, m_{x,i}, s_{x,i}) = \frac{1}{\sqrt{2\pi s_{x,i}}} e^{-\frac{(e_x - m_{x,i})^2}{2s_{x,i}}}, \quad i = 1, \dots, 3, \quad (6.4)$$

$$\mathcal{G}^j(t_s, m_{s,j}, s_{s,j}) = \frac{1}{\sqrt{2\pi s_{s,j}}} e^{-\frac{(t_s - m_{s,j})^2}{2s_{s,j}}}, \quad j = 1, \dots, 3, \quad (6.5)$$

where m_x, m_s are the centres of Gaussian functions while s_x and s_s define their dispersion. Having the membership functions, let us define the normalized rule firing strength:

$$\mu_{i,j} = \frac{\mathcal{G}^i(e_x, m_{x,i}, s_{x,i})\mathcal{G}^j(t_s, m_{s,j}, s_{s,j})}{\mathcal{G}_t}, \quad i, j = 1, \dots, 3, \quad (6.6)$$

where

$$\mathcal{G}_t = \sum_{i=1}^3 \sum_{j=1}^3 \mathcal{G}^i(e_x, m_{x,i}, s_{x,i})\mathcal{G}^j(t_s, m_{s,j}, s_{s,j}). \quad (6.7)$$

Finally, the model (6.3) can be transformed into

$$c_m = \sum_{i=1}^3 \sum_{j=1}^3 \mu_{i,j} \left(p_{1,m}^{i,j} n_c + p_{2,m}^{i,j} m_p + p_{3,m}^{i,j} C_c \right). \quad (6.8)$$

As a result, the TS model can be expressed in the following form:

$$c_m = \sum_{i=1}^3 \sum_{j=1}^3 \mu_{i,j} r_m^T p_m^{i,j}, \quad (6.9)$$

where $r_m = [n_c, m_p, C_c]^T$ and $p_m^{i,j} = [p_{1,m}^{i,j}, p_{2,m}^{i,j}, p_{3,m}^{i,j}]^T$ are the regressor and parameter vectors, respectively. The final step boils down transforming (6.9) to the regressor form:

$$c_m = r_m^T p_m, \quad (6.10)$$

where

- $r_m = [\mu_{1,1} r_m^T, \mu_{1,2} r_m^T, \dots, \mu_{2,2} r_m^T]^T$,
- $p_m = [(p_m^{1,1})^T, (p_m^{1,2})^T, \dots, (p_m^{3,3})^T]^T$.

Using a similar line of reasoning, the TS model (6.2) can be derived:

$$c_l = r_l^T p_l, \quad (6.11)$$

where

- $r_l = [\mu_{1,1} r_l^T, \mu_{1,2} r_l^{1,2} T, \dots, \mu_{3,3} r_l^T]^T$,
- $p_l = [(p_l^{1,1})^T, (p_l^{1,2})^T, \dots, (p_l^{3,3})^T]^T$.

with $r_l = n_c$.

One important feature related to the models (6.10)–(6.11) is that they are linear with respect to p_m and p_l . This appealing feature makes it possible to use the celebrated least-square algorithm [12, 13] for estimating these parameters.

The objective of the remaining part of this section is to provide a concise outline of the general algorithm for designing (6.10)–(6.11):

KIS.BOX assignment: assign workers with different experience (see Table 6.3) to different KIS.BOXes.

Production plan: determine a strategy for gathering representative and possibly large data sets concerning workers with different experience that operate according to the battery production plan.

Data gathering: perform the data gathering procedure:

- obtain the data from the Data trend chart;
- synchronize the obtained data with the production plan;
- based on the KIS.BOX assignment, extend the data set with worker experience.

Model design: use the recursive least square algorithm to determine (6.10)–(6.11).

> Highlights

The model (6.10)–(6.11) may have several prospective applications. Having a manufacturing plan corresponding to the required number of batteries as well as group of workers performing within the shifts, one can use (6.10)–(6.11) to

1. schedule production and estimate its feasibility;
2. schedule the work of individual workers, i.e., the frequency of delivering components;
3. make the production process transparent, and hence increase various KPIs, e.g., OEE;
4. use it as a digital twin of the worker, which can be employed for their training and continuous skill improvements.

6.2 Transportation Process

The objective of this section is to present an approach for modelling forklift operator performance within a warehouse, which has been recently developed by the authors [1, 14]. The proposed framework is designed for a fleet of cooperating forklifts working within a high storage warehouse. A sample part of such a system is presented in Figs. 6.3–6.4. As can be observed, KIS.Devices are used as communication tools. Thus, each forklift is equipped with KIS.BOX while KIS.LIGHTs are installed on each transfer station. The transfer station is defined as a place for storing an item (product, materials, etc.), which has been delivered to it via different transportation means, e.g., automated guided vehicles [15, 16].

Let us start with stating that an item associated with the k -th transportation event is identified by

$$\mathbb{L}(k) = \{s(k), t(k), d(k), m(k), q(k)\}, \quad (6.12)$$

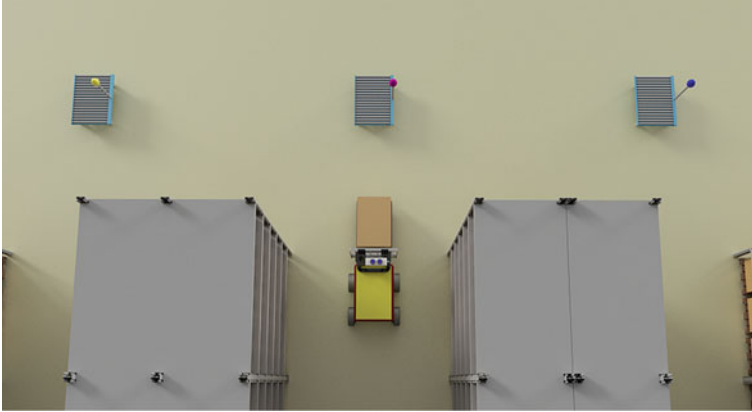


Fig. 6.3 Sample part of the warehouse shopfloor (top view)



Fig. 6.4 Sample part of the warehouse shopfloor (side view)

where

- $s(k)$ stands for the storage place identifier in a warehouse;
- $t(k)$ is the transfer station identification number;
- $d(k)$ is the distance (in meters) between the places $t(k)$ and $s(k)$;
- $m(k)$ signifies the item mass (in kg);
- $q(k)$ denotes the item volume (in m^3).

The objective of the remaining part of this chapter is to provide a concise procedure for designing a forklift driver performance model using KIS.ME. Let \mathbb{P} denote the set of all require transportation events, which is defined as follows:

$$\mathbb{P} = \{\mathbb{L}(0), \mathbb{L}(1), \dots, \mathbb{L}(n_E)\}, \tag{6.13}$$

where n_E signifies the number of transportation events.

> Realization constraints

For the sake of simplicity, the following is assumed:

- The transfer stations are situated close to the entrance of each aisle (see Fig. 6.3).
- Each forklift is operating in a single aisle.
- An item designated to a given aisle can be delivered to its direct or closest neighboring transfer stations (see the illustrative example given in Fig. 6.3).
- Each KIS.LIGHT operational LED illuminates in a different color, and hence it uniquely identifies the transfer station, i.e., the transfer station identifier $t(k)$ is equal to the numerical value of its color (see Table 2.2).
- KIS.LIGHT digital inputs uniquely define the designated aisle of an item (see Table 6.4), which can be in front of it or on the left or right hand side. It is assumed that the states of digital inputs are set by the transportation means which delivers an item to the transportation station. Note also that if an item is collected by a forklift, then the digital inputs are set to the “None” state (see Table 6.4), which signifies the fact that the transfer station is empty.
- it is assumed that there exists a transportation system which cooperates with KIS.ME and delivers items to the appropriate transfer stations;
- the availability of a forklift is signified by the facts that of the both KIS.BOX Button 1 and 2 the operational LED colors are black, i.e., they do not illuminate.
- The item can be released at a transfer station iff it is empty.

Thus, transportation from the transfer stations to the storage places is realized as follows:

- Step 0: Set $k = 1$;
- Step 1: The k -th item is delivered to $t(k)$, which is identified by $\mathbb{L}(k)$.
- Step 2: Using the $t(k)$ transfer station’s KIS.LIGHT operational LED color (numerical value $t(k)$) and digital inputs states (Table 6.4), KIS.MANGER determines the designated aisle, i.e., a forklift and its associated KIS.BOX.
- Step 3: If of the both KIS.BOX Button 1 and Button 2 the operational LED colors are black, then their color is changed to the one with the numerical value equal to $t(k)$ (blue in Fig. 6.3).
- Step 4: The forklift operator presses KIS.BOX Button 1 and its operational LED starts lighting in red. The operator starts the $\mathbb{L}(k)$ transportation event by collecting the k -th item and going to the storage place $s(k)$.

Table 6.4 Aisle assignment to the transfer station

Aisle	KIS.LIGHT digital input 1	KIS.LIGHT digital input 2
In front	1	1
Left hand side	1	0
Right hand side	0	1
None	0	0

- Step 5: The forklift operator arrives at $s(k)$, pushes KIS.BOX Button 1 again and then its operational LED changes its color to black.
- Step 6: The forklift operator presses KIS.BOX Button 2 and its operational LED starts lighting in red. The operator releases the item at $s(k)$ and moves to the beginning of the aisle.
- Step 7: The operator arrives at the beginning of the aisle, presses KIS.BOX Button 2 and its operation LED changes its color to black.
- Step 8: If $k < n_E$, then set $k = k + 1$, else STOP.

Note that if the condition of Step 3 is not satisfied, then KIS.MANGER Rule engine waits until it is feasible. Moreover, the black and red colors are used for the task identification purpose, and hence they cannot be used for the identification of the transfer stations. Finally, the objective of this data acquisition is to collect measurements associated with

- Steps 4–5: the duration $f(k)^m$ in which the operational LED color of the i -th forklift KIS.BOX Button 1 is red;
- Steps 6–7: the duration $b(k)^m$ in which the operational LED color of the i -th forklift KIS.BOX Button 2 is red.

As in the previous section, the data concerning the performance of the above system can be gathered using the Data trend chart. These data include a timestamp which clearly identifies the working time within the shift. Subsequently, the obtained data can be merged with the detailed description of all items (6.13) as well as operator experience, which is associated with the KIS.BOX being used. As a result, $k = 0, \dots, n_E$ Datapoints are obtained, which can be used for designing a model of the forklift operator's performance:

$$L_f(k) = \{t(k), s(k), d(k), m(k), q(k), f(k)^m, b(k)^m, e_x(k), t_s(k)\}, \quad (6.14)$$

where $e_x(k)$ and $t_s(k)$ are respectively experience and working time within the shift of the forklift operator transporting the k -th item.

Similarly as in the preceding section, two TS models will be developed. The fuzzy model is divided into two sub-models:

$$f = f_f(d, m, q, e_x, t_s), \quad (6.15)$$

$$b = f_b(d, m, q, e_x, t_s). \quad (6.16)$$

Finally, $f_f(\cdot)$ and $f_b(\cdot)$ signify a given model structure used for calculating either $f(k)$ or $b(k)$ for the k -th item. Owing to safety purposes, the maximum velocity limit can be imposed, e.g., at the level of $v_f = 5 \text{ km/h} = 1.39 \text{ m/s}$:

$$f \geq \frac{1}{v_f}d, \quad (6.17)$$

which is implied by the fact that $distance = time \times velocity$. The same constraint is imposed on b , i.e., $f \geq \frac{1}{v_f}d$. It is also evident that the mass m and volume q of an item have a direct impact on f .

Similarly as in the preceding section, two fuzzy premise variables are introduced, which are detailed in Table 6.3. Under these preliminaries, the TS model of (6.15) is

$$\begin{aligned} & \text{IF } t_s \in \mathbb{T}_{s,j} \text{ and } e_x \in \mathbb{E}_{x,i}, \text{ THEN} \\ & f = p_{1,f}^{i,j}d + p_{2,f}^{i,j}m + p_{3,f}^{i,j}q, \quad i, j = 1, \dots, 3, \end{aligned} \quad (6.18)$$

while (6.16) obeys

$$\begin{aligned} & \text{IF } t_s \in \mathbb{T}_{s,j} \text{ and } e_x \in \mathbb{E}_{x,i}, \text{ THEN} \\ & b = p_{1,b}^{i,j}d + p_{2,b}^{i,j}m + p_{3,b}^{i,j}q, \quad i, j = 1, \dots, 3, \end{aligned} \quad (6.19)$$

where p_i are the model parameters which have to be determined while $\mathbb{E}_{x,j}$ and $\mathbb{T}_{s,i}$ are fuzzy sets associated with Table 6.3. Using the rule firing strength (6.6), the models (6.18)–(6.19) can be written as follows:

$$f = \sum_{i=1}^3 \sum_{j=1}^3 \mu_{i,j} \left(p_{1,f}^{i,j}d + p_{2,f}^{i,j}m + p_{3,f}^{i,j}q \right), \quad (6.20)$$

$$b = \sum_{i=1}^3 \sum_{j=1}^3 \mu_{i,j} \left(p_{1,b}^{i,j}d + p_{2,b}^{i,j}m + p_{3,b}^{i,j}q \right). \quad (6.21)$$

The final set boils down to transforming (6.20)–(6.21) to the regressor form:

$$f = r_f^T p_f, \quad (6.22)$$

where

- $r_f = [\mu_{1,1}r_f^T, \mu_{1,2}r_f^T, \dots, \mu_{2,2}r_f^T]^T$,
- $p_f = [(p_f^{1,1})^T, (p_f^{1,2})^T, \dots, (p_m^{3,3})^T]^T$,

and $r_f = [d, m, q]^T$. The model (6.21) can be formulated in a similar way:

$$b = r_b^T p_b, \quad (6.23)$$

where

- $r_b = [\mu_{1,1}r_b^T, \mu_{1,2}r_b^T, \dots, \mu_{2,2}r_b^T]^T$,
- $p_b = [(p_b^{1,1})^T, (p_b^{1,2})^T, \dots, (p_b^{3,3})^T]^T$.

The objective of the remaining part of this section is to provide a concise outline of the general algorithm for designing (6.22)–(6.23):

KIS.BOX assignment: Assign forklifts operators with different experience (see Table 6.3) to different KIS.BOXes.

Transportation plan: Obtain the transportation plan (6.13) consisting of n_E transportation events.

Data gathering: Perform the data gathering procedure according to the above 8-step strategy:

- obtain the data from the Data trend chart;
- synchronize the obtained data with the transportation plan;
- based on the KIS.BOX assignment, extend the data set with the forklift operator's experience.

Model design: Use the recursive least square algorithm to determine (6.22)–(6.23).

> Highlights

The model (6.22)–(6.23) may have several crucial applications. Having a transportation plan (6.13), one can use (6.22)–(6.23) to

1. schedule transportation of items to the transfer stations and estimate its feasibility;
2. schedule the work of individual forklift operators, which will minimize a possible bottle neck effect at the transfer stations;
3. make the transportation process transparent, and hence increase various KPIs like the ones proposed in Sect. 5.1;
4. use it as a digital twin of the forklift operators, which can be employed for their training and permanent skill improvement.

6.3 Integrating Workers Within a Semi-automatic Assembly System

A sample general semi-automatic assembly scheme is presented in Fig. 6.2. Indeed, it may be surrounded by conveyor belts or other transportation means that provide suitable components for the assembly process. Let us begin with recalling the fact that the worker can start mounting the next battery iff the previous one is completed. Let us also recall (see Sect. 6.1.1) that the completion time is the sum of

1. battery cell mounting time c_m ;
2. cell-controller linking time c_l .

Let us define the following variables (cf. Table 6.2 and Fig. 6.2):

- $x(k)$: start time of mounting the k -th battery,
- $u_m(k)$: cell delivery time for the k -th battery,
- $u_c(k)$: controller delivery time for the k -th battery.

This nomenclature allows formulating a simple formal description pertaining to the evolution of the start time of mounting consecutive batteries:

$$x(k+1) = \max(x(k) + c_m(k) + c_l(k), u_m(k+1), u_c(k+1)), \quad (6.24)$$

where $c_m(k)$, $c_l(k)$ are respectively c_m and c_l for the k -th battery. This means that the maximum of the times included in (6.24), i.e., in $\max(\cdot)$, determines the start time of mounting the $k+1$ st battery.

Checking manufacturing feasibility

Without any loss of generality, let us assume that the starting assembly time is $x(0) = 0$, which may correspond to 8.00 h. Moreover, let the battery cells and the controller be available at that time, i.e., $u_c(0) = 0$ and $u_l(0) = 0$.

There is a set of 10 different batteries to be mounted. Having the worker experience, as well as required parameters of the batteries described along with (6.1)–(6.2) and the performance model (6.10)–(6.11), one can calculate the expected battery mounting times $c_m(k) + c_l(k)$, which are given in Table 6.5. The cells and the controller are delivered to the assembly station every 19 and 20 min, respectively. The problem is to check if there is a chance to mount all 10 batteries in 200 min. For that purpose, one can use (6.24). As an example let us employ it for calculating $x(1)$, i.e., the start time of mounting battery number 1 (Table 6.6):

$$x(1) = \max(x(0) + c_m(0) + c_l(0), u_c(1), u_l(1)) = \max(23, 19, 20) = 23. \quad (6.25)$$

As a result, the mounting start time for the last battery is $x(9) = 184$ while its expected assembly time is 20 min, and hence all batteries will be completed in 204 min.

Table 6.5 Expected battery mounting times

k -th Battery	$c_m(k) + c_l(k)$
0	23
1	19
3	16
4	20
5	15
6	27
7	12
8	20
9	24

Table 6.6 Expected battery mounting times

k -th Battery	$c_m(k) + c_l(k)$	$x(k)$	$u_c(k)$	$u_l(k)$
0	23	0	0	0
1	19	23	19	20
2	16	42	38	40
3	20	60	57	60
4	15	80	76	80
5	27	100	95	100
6	12	127	114	120
7	20	140	133	140
8	24	160	152	160
9	20	184	171	180

➤ Towards flexible manufacturing

A direct calculation of the total battery assembly time,

$$T_{\text{total}} = \sum_{k=0}^9 (c_m(k) + c_l(k)) = 196 \text{ [min]}, \quad (6.26)$$

clearly indicates that it is possible to perform the above task in 200 min. This, however, requires a different strategy for calculating component delivery times $u_m(k)$ and $u_c(k)$. Indeed, there are different types of batteries with different expected assembly times (cf. Table 6.5). Thus, because the employed delivery rate is constant, it prevents attaining optimal assembly performance.

To settle this important issue, it is proposed to extend the scheme presented in Fig. 6.2 with an external device and scheduling software. As a result, the scheme shown in Fig. 6.5 is obtained. The purpose of the external software is to calculate optimal component delivery times $u_m(k)$ and $u_c(k)$, which will make it possible to have the components just on time. The software should also take into account inevitable delays which can be caused by the worker. Additionally, such delays should be compensated for as much as possible. For further details the reader is referred to the authors' publications concerning such developments [11, 16, 17]. Indeed, using the fault-tolerance concept, these approaches can also minimize various unappealing phenomena like, e.g., delays. Additionally, the control process involves a cost function, which can take into account a wide range of economic aspects [14].

As can be seen in Fig. 6.5, the hardware is communicated with KIS.BOX through its digital outputs, which provide information about the current assembly status. Finally, the scheduling sequence is employed to control the item delivery process.

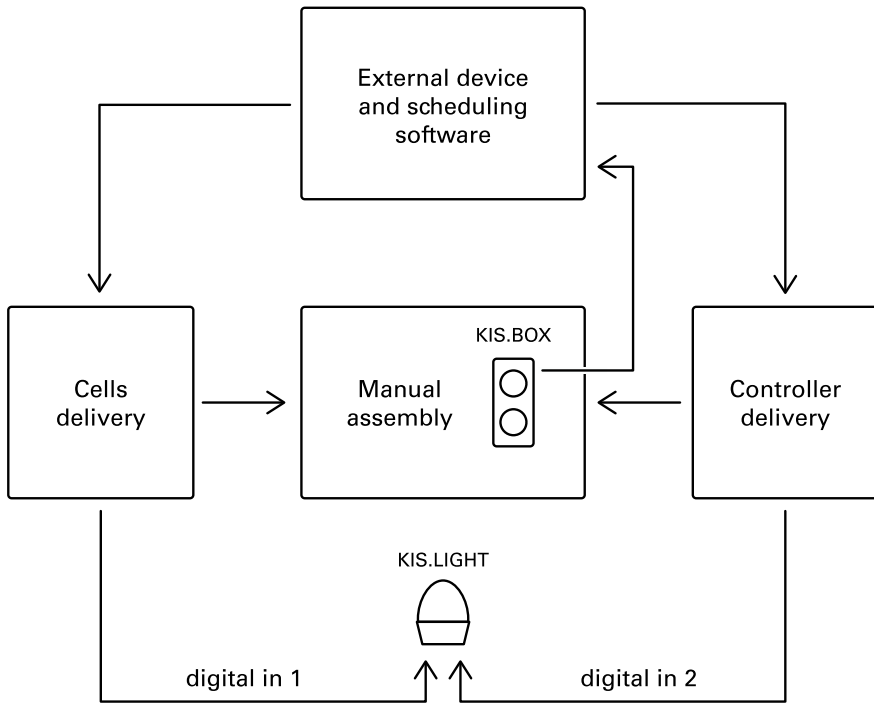


Fig. 6.5 General overview of the manual assembly station with the KIS.ME infrastructure as well as external hardware and software

6.4 Scheduling Transportation Actions

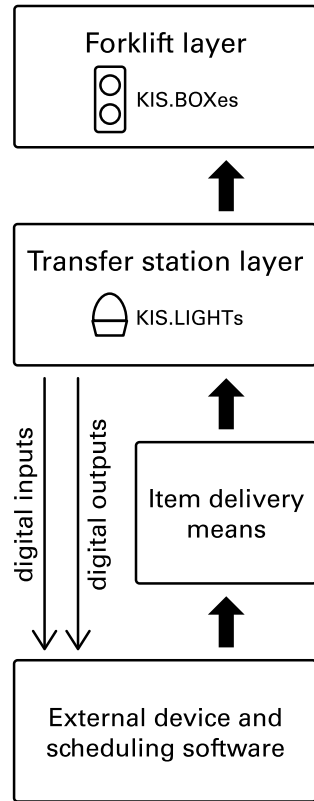
In Chap. 3 and Sect. 5.1, a set of logistic and transportation solutions were proposed along with measures that can be used for assessing their performance. However, in all cases the decisions concerning the required transportation actions were made either by the transporter operator or the KIS.MANAGER operator. In this section, the warehouse transportation system proposed in Sect. 6.2 is extended with a scheduling framework, which makes it possible to deliver the items (cf. Fig. 6.3) just on time. Similarly as in Sect. 6.3, let us introduce the following variables concerning the i -th forklift:

- $x_i(k)$: the start time of collecting the k -th item from the $T(k)$ transfer station,
- $u(k)$: the delivery time of the k -th item to the $T(k)$ transfer station,
- $v_i(k)$: a decision variable associating the k -th item with the i -th forklift.

The decision variable may have two states:

- $v_i(k) = 1$: the k -th item is transported by the i -th forklift,
- $v_i(k) = 0$: the k -th item is not transported by the i -th forklift.

Fig. 6.6 General overview of the transportation system with the KIS.ME infrastructure as well as the external hardware and software



A scheme of the proposed general transportation framework is presented in Fig. 6.6. From the KIS.MANAGER viewpoint, the proposed scheme extends the one discussed in Sect. 6.2 by introducing rules of the following form: If a forklift performed transportation of the k -th item, then the false–true–false sequence is sent to KIS.LIGHT digital output 1 associated with the $T(k)$ transfer station. This slight modification allows gathering measurements pertaining to the transportation time of the k -th item. Indeed, this process starts from the “None” state (see Table 6.4) of the KIS.LIGHT associated with the $T(k)$ transfer station and ends with the false–true–false sequence on its digital output 1.

Under the above preliminaries, the time-driven model of the i -th forklift is given by

$$x_i(k + 1) = \max (x_i(k) + f_{d,i}(x) + b_{d,i}(k), u(k + 1)v_i(k + 1)), \quad (6.27)$$

where

$$f_{d,i}(k) = f_i(k)v_i(k), \quad (6.28)$$

$$b_{d,i}(k) = b_i(k)v_i(k), \quad (6.29)$$

while $f(k)$ and $b(k)$ can be calculated using the model (6.22)–(6.23) developed in Sect. 6.2. This means that, depending on the decision $v_i(k)$, the variables $f_{d,i}(k)$ and $b_{d,i}(k)$ can be equal to either 0 or $f_i(k)$ and $b_i(k)$, respectively.

Two forklifts example

Let us consider two forklifts and one transfer station located in front of the aisle corresponding to the second forklift. Figure 6.3 shows the second forklift and the transfer station with the KIS.LIGHT operational LED lighting in magenta. The first forklift operates in front of the transfer station with the KIS.LIGHT operational LED lighting in yellow. Let us imagine that the item is delivered to the transfer station (magenta) and it has to be transported by the first forklift. This means that the corresponding KIS.LIGHT digital input is set to the “Left hand side” state (cf. Table 6.4). Based on such information, the external software should set the decision variables in the following way: $v_1(0) = 1$ and $v_2(0) = 0$, which clearly indicates that the first forklift has to perform transportation of the 0th item. Let us also assume that $x_1(0) = 0$ and $x_2(0) = 0$, which may correspond to 8.00, i.e., the beginning of the shift. Moreover, let $f_1(0) = 3$ and $b_1(0) = 2$ minutes. Thus, Eqs. (6.27)–(6.29) imply that

$$x_1(1) = \max(0 + 3 + 2, u(1)v_1(k)) = \max(5, u(1)v_1(1)), \quad (6.30)$$

$$x_2(1) = \max(0, u(1)v_2(1)) = u(1)v_2(1). \quad (6.31)$$

From the above results, it is evident that, if the subsequent, i.e., $k = 1$, item should be transported by the first forklift ($v_1(1) = 1$ and $v_2(1) = 0$), then $u(1) \leq 5$ guarantees just-on-time performance. Contrarily, if the subsequent, i.e., $k = 1$, item should be transported by the first forklift ($v_1(1) = 0$ and $v_2(1) = 1$), then $u(1) = 0$ guarantees just-on-time performance.

The above preliminary results formed the basis for formulating the scheduling strategies proposed by the authors [1, 11, 14, 15, 17], which can be efficiently used for various kinds of problems. The crucial element of these strategies is the cost function:

$$J = \sum_{k=0}^{n_E-1} u(k), \quad (6.32)$$

which has to be maximized for all n_E transportation events (6.13) taking into account (6.27) and the related constraints [1, 11, 14, 15, 17]. The process of maximizing (6.32) can be interpreted as finding the just-on-time item delivery times $u(k)$ for the forklift transportation system.

6.4.1 Health-aware and Fault-Tolerant Transportation Scheduling

The scheduling software presented in Fig. 6.6 can be extended with additional elements, which can settle the following problems:

- The proposed transportation system is a time-driven one, i.e., all modelled and optimized variables represent certain operation times. From the classical system control viewpoint [12, 18], the fault is defined as an unpermitted deviation of at least one characterised property of the system compared to its nominal value. Thus, delays within transportation system can be interpreted as faults [11, 14–17]. This justifies the need for implementing fault-tolerance in the transportation system.
- The application of indoor-operating forklifts makes it necessary to use various kinds of batteries or accumulators. Thus, while scheduling the work of forklifts one has to take into account a compulsory recharging process as well as the state of charge (SOC) and the state of health (SOH) of the batteries.

Before settling the above issues, it is necessary to provide a list of crucial constraints which influence the performance of the transportation system [14]:

Transfer constraint: For the same transfer station, it is possible to deliver a new item iff the previous one has been removed. Thus, for all l and k ,

$$\text{IF } l > k \text{ and } T(k) = T(l) \text{ THEN } u(k) \leq u(l) + t(j), j < n_E, \quad (6.33)$$

where $t(j) > 0$ is a unknown time between delivering the k -th and l -th items to the $T(k) = T(l)$ transfer station.

Scheduling constraint: This constraint simply states that all n_E starting times of transportation events should be bounded:

$$u(k) \leq u_E, \quad (6.34)$$

where $u_E > 0$ is maximum scheduled time for starting all n_E transportation events.

Energy constraint: It is assumed that each forklift is equipped with the battery management system (BMS), which provides information about the maximum operational time of the i -th forklift $\tau_i(k)$ for the k -th transportation system.

Let us start with analysing potential faulty situations, which will have direct influence on the feasibility of constraints. The delay in taking the k -th item will cause $t(j)$ to be extended, i.e., the delivery of the subsequent l -th item will be delayed. Thus, with

appropriately large $t(j) > 0$, the constraint (6.33) is always feasible. Contrarily, a faulty situation may cause (6.34) to be infeasible. Thus, fault-tolerance can be attained by introducing a relaxation variable $\bar{u} \geq 0$, which should be as small as possible. Additionally, if the faulty situation makes the measured transportation times larger than the nominal values, then they should replace them in the model (6.27) of the i -th forklift, which suffers from the delay. Thus, to achieve fault-tolerance, the minimized cost function should be given as follows:

$$J = -p_1 \sum_{k=0}^{n_E-1} u(k) + p_2 \bar{u}, \quad (6.35)$$

where $p_1 \geq 0$ and $p_2 \geq 0$, $p_1 + p_2 = 1$ are weighting parameters, which can be used for attaining a balance between fault-tolerance and performance.

Let us proceed to health-aware analysis. For simplicity, it is assumed that each forklift is operating in a single aisle. However, more flexibility can be attained by removing this constraint. Thus, one can imagine that the tasks with shorter (smaller $f_i(k) + b_i(k)$) and less energy-demanding transportation tasks are realized by forklifts with smaller $\tau_i(k)$. This can be formally expressed by the following cost function:

$$J_H = - \sum_{k=0}^{n_E-1} \sum_{i=1}^{n_F} (\tau_i(k) - b_i(k) - c_i(k)), \quad (6.36)$$

where n_F is the number of forklifts. Finally, the minimized cost function takes the form

$$J = -p_1 \sum_{k=0}^{n_E-1} u(k) + p_2 \bar{u} + p_3 J_H, \quad (6.37)$$

where $p_3 \geq 0$, $p_1 + p_2 + p_3 = 1$.

➤ Economy oriented cost function

The cost function (6.37) is very general and can be extended with additional components corresponding to the operational cost of the transportation system. This is especially important in the case of a fleet consisting of different forklifts. Thus, each forklift may involve different operational costs.

6.5 Training Exercise: Work Scheduling

6.1 Scheduling tasks of three workers

Exercise requirements: The exercise requires access to one KIS.LIGHT, three KIS.BOXes and a photoelectric sensor with an M12 8 pin interface (see Sect. 3.1).

1. There are three workers that can perform the same kind of tasks using the component sets provided by the conveyor belt (see Fig. 6.7).
2. The conveyor belt transfers a component set to the beginning of the working area and then stops.
3. Use the photoelectric sensor to detect the arrival of the component set and feed this information to KIS.LIGHT digital input 1.
4. Each worker operates according to the states presented in Table 6.7.
5. Each worker has its own individual KIS.BOX. Implement the state-space model (6.7) for each KIS.BOX (see Sect. 2.10). As a trigger use an action associated with pressing KIS.BOX Button.
6. Each worker can be either present or absent at the workplace. Implement the state-space model (Table 6.8) for each KIS.BOX. As a trigger use an action associated with pressing KIS.BOX Button 2.
7. Depending on the KIS.LIGH digital outputs' state (see Table 6.9), the conveyor belt can move a component set to one of the three workplaces.
8. using Rule engine Implement the system logic in such a way that new component sets are delivered to the workers being present and in an idle state.

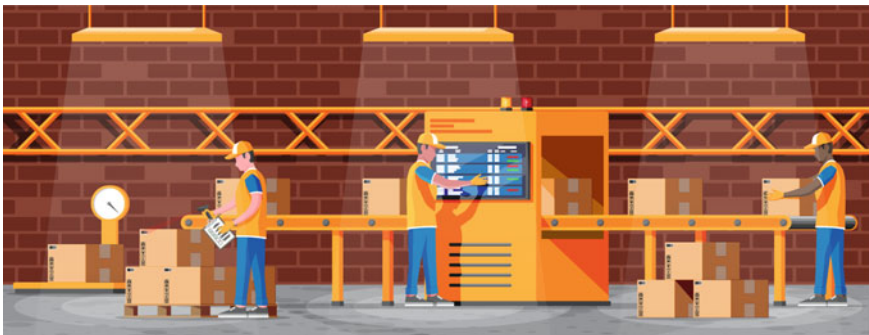


Fig. 6.7 Scheduling tasks of three workers

Table 6.7 Worker's KIS.BOX Button 1 operational LED states

State	KIS.BOX Button 1 operational LED color
Idle	Green
Mounting	Red
Packing	Yellow

Table 6.8 Worker's KIS.BOX Button 2 operational LED states

State	KIS.BOX Button 1 operational LED color
Present	Green
Absent	Red

Table 6.9 KIS.LIGHT transportation actions

Worker	KL digital output 1	KL digital output 2
1	0	1
2	1	0
3	1	1

6.6 Concluding Remarks

The main objective of this chapter was to provide a selected list of motivating examples pertaining to potential applications of KIS.ME. Although the presented instances concern a dedicated application study, they can be relatively easily adapted to different applications frequently encountered in various industries. In particular, it was shown how to model human operator performance both in assembly and transportation tasks. The resulting model is particularly important for various kinds of planning and scheduling. Indeed, such models make it possible to obtain transparency and predictability of the controlled and monitored system. Additionally, as most indoor transportation means employ various kinds of accumulators, it is profitable to take into account their health. Thus, it was shown how to include such information in the scheduling procedure. As a result, the overall accumulator health of the transporter fleet can be balanced. Another important aspect is that the human operated system may suffer from various kinds of delays. Thus, it was shown how to attain fault-tolerance which can prevent such a kind of unappealing situations. Finally, the chapter ended with advanced training exercises, which pertain to work scheduling of three workers operating a conveyor belt.

References

1. M. Witczak, B. Lipiec, M. Mrugalski, L. Seybold, Z. Banaszak. Fuzzy modelling and robust fault-tolerant scheduling of cooperating forklifts, in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (IEEE, Glasgow, 2020), pp. 1–10
2. L.A. Zadeh, Fuzzy logic. *Computer* **21**(4), 83–93 (1988)
3. G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic* (Prentice Hall, New Jersey, 1995), vol. 4
4. P. Hájek, *Metamathematics of Fuzzy Logic* (Springer Science & Business Media, Berlin, 2013)
5. Z. Nivolianitou, M. Konstantinidou, *A fuzzy modeling application for human reliability analysis in the process industry, in Human Factors and Reliability Engineering for Safety and Security in Critical Infrastructures* (Springer, Berlin, 2018), pp.109–154

6. M. Konstandinidou, Z. Nivolianitou, Ch. Kiranoudis, N. Markatos, A fuzzy modeling application of CREAM methodology for human reliability analysis. *Reliab. Eng. Syst. Saf.* **91**(6), 706–716 (2006)
7. Q. Zhou, Y.D. Wong, H.S. Loh, K.F. Yuen, A fuzzy and bayesian network CREAM model for human reliability analysis-the case of tanker shipping. *Saf. Sci.* **105**(6), 149–157 (2018)
8. R. Sujatha, V. Sharmila, P. Hema, M. Amruth Varshinee, A. Prasnath. Some aspects of human error analysis using fuzzy relations. *Int. J. Pure Appl. Math.* **117**(21), 311–325 (2017)
9. M.A. Segura, S. Hennequin, B. Finel, Human factor modelled by fuzzy logic in preventive maintenance actions. *Int. J. Operat. Res.* **27**(1–2), 316–340 (2016)
10. G. Suprakash, K. Pramod, Y.R. Gunda, A fuzzy causal relational mapping and rough set-based model for context-specific human error rate estimation. *Int. J. Occup. Saf. Ergon.* **27**(1), 63–78 (2021)
11. L. Seybold, M. Witczak, P. Majdzik, R. Stetter, Towards robust predictive fault-tolerant control for a battery assembly system. *Int. J. Appl. Math. Comput. Sci.* **25**(4), 849–862 (2015)
12. M. Witczak, *Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems: Analytical and Soft Computing approaches* (Springer International Publishing, Heidelberg, Germany, 2014)
13. O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes* (Springer Nature, Berlin, 2020)
14. M. Witczak, Lo. Seybold, G. Bocewicz, M. Mrugalski, A. Gola, Z. Banaszak. A fuzzy logic approach to remaining useful life control and scheduling of cooperating forklifts, in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (IEEE, Luxemburg, 2021), pp. 1–8
15. M. Witczak, P. Majdzik, R. Stetter, B. Lipiec, A fault-tolerant control strategy for multiple automated guided vehicles. *J. Manuf. Syst.* **55**(4), 56–68 (2020)
16. P. Majdzik, M. Witczak, B. Lipiec, Z. Banaszak. Integrated fault-tolerant control of assembly and automated guided vehicle-based transportation layers. *Int. J. Comput. Integr. Manuf.* 1–18 (2021)
17. P. Majdzik, A. Akielaszek-Witczak, L. Seybold, R. Stetter, B. Mrugalska, A fault-tolerant approach to the control of a battery assembly system. *Control Eng. Pract.* **55**(10), 139–148 (2016)
18. M. Witczak, *Modelling and Estimation Strategies for Fault Diagnosis of Non-linear Systems* (Springer, Berlin, 2007)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

