Integrated minimum-time trajectory generation, fault detection, and reconfiguration for a double-tank system using flatness and B-splines

Fajar Suryawan, José De Doná, and María Seron

Abstract— **The main contribution of this paper is to provide a unified treatment to the problems of constrained minimum-time trajectory generation, fault detection and identification, and (after a fault has been detected and identified) trajectory reconfiguration, in an integrated scheme using a differential flatness and B-splines parameterisation. Using the flatness/B-splines parameterisation the problem of minimum-time constrained trajectory planning is cast into a feasibility-search problem in the splines control-points space, in which the constraint region is characterised by a polytope. A close approximation of the minimum-time trajectory is obtained by systematically searching the end-time that makes the constraint polytope to be minimally feasible. Fault detection is carried out by using B-splines in an FIR filter implementation. Thus, the three—traditionally dealt with separately—problems (namely, trajectory generation, fault detection, and trajectory reconfiguration) are solved in a unified manner, using the same mathematical/computational tools. This, not only offers an elegant solution, but also has the potential to simplify the coding of the algorithms for the real-time application of the strategy. All through the paper, a case-study consisting in an input-constrained double-tank system is analysed in order to illustrate the techniques in an intuitive manner.**

I. Introduction

In control systems, a desirable strategy is often to perform a task as fast as possible, considering all constraints. This problem, often referred to as time-optimal control or minimum-time control, has been a long standing problem in the systems and control literature, as well as in applied mathematics. The problem can be traced back to, e.g., the work of Bellman et. al. [1]. Despite the inherently interesting nature of the problem, analytical solutions are often very complex, even for low dimensional linear systems. In this regard, there are only very few treatments in the literature dealing with relatively complex problems.

The approach to minimum-time control illustrated here stems from our previous work [2], [3], where, using differential flatness and B-splines, every signal and constraint are mapped to the *control-point space* of B-splines. The constraints form a polytope in this space whose shape changes as the end-time of the parameterisation is varied. This fact is exploited to search for a polytope that is minimally feasible, at which point a minimum time is reached. A preliminary version of this method (for SISO systems) can be found in a recent conference paper by the authors [4].

The authors are with Centre for Complex Dynamic Systems and Control (CDSC), University of Newcastle, Australia
Fajar.Suryawan@uon.edu.au
Jose.DeDona@newcastle.edu.au
Maria.Seron@newcastle.edu.au

It is well known that for linear systems constrained only on the input, the resulting time-optimal control solution is bang-bang. For these systems, the method proposed here is sub-optimal compared to bang-bang control. However, advantages of the method include: 1) the ability to specify initial and final conditions, including the derivatives, of the inputs, states, and outputs, 2) the ability to naturally deal with constraints on inputs, states, and outputs, including their derivatives, 3) the ability to naturally deal with non-minimum phase and unstable systems, 4) there are no intersampling issues (since no discretisation is involved), 5) the signals produced are smoother due to the use of splines, 6) the method can be naturally extended to MIMO systems.

This paper also considers, in an integrated fashion with the above mentioned minimum-time trajectory generation, the use of differential flatness and B-splines for fault-detection. For flat linear systems, all signals can be expressed as linear combinations of the flat outputs and their derivatives [5]–[7]. Hence, in normal conditions the actual signals and the signals constructed from the flat outputs should be equal up to the effect of noises and model uncertainties. To process the flat outputs to estimate all the other signals, B-Spline tools are used in this paper. This method has been successfully applied to a laboratory-scale magnetic levitation system and reported recently in [8].

After the occurrence of a significant fault, it is necessary to accommodate the fault, either by gracefully shutting the plant down, or by reconfiguring the control strategy. The main contribution of this paper is that it provides a unified treatment to the problems of constrained minimum-time trajectory generation, fault detection and identification, and trajectory reconfiguration, in an integrated scheme using differential flatness and B-splines parameterisations. Thus the three problems—traditionally dealt with separately—are solved in a unified manner, using the same mathematical/computational tools. This, not only offers an elegant solution, but also has the potential to simplify the coding of the algorithms for the real-time application of the strategy. This integrated scheme is applied to a case-study consisting in the total loss of an actuator in a MIMO system; namely, a double-tank system with input pumps in both tanks and the tank levels as the outputs.

II. Plant description

A. Two-inputs, two-outputs model

Consider the following linearised model of a two-inputs, two-outputs, double-tank system around an equilibrium point

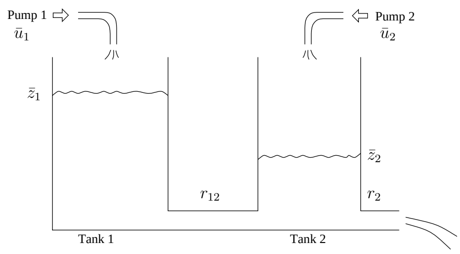


Fig. 1. Diagram of the double tank system.

(see Fig. 1)

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} = A\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}B_1 & B_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} \tag{1}$$

where

$$A = \begin{bmatrix}-\frac{1}{r_{12}} & \frac{1}{r_{12}}\\ \frac{1}{r_{12}} & -\left(\frac{1}{r_{12}}+\frac{1}{r_2}\right)\end{bmatrix},\ B_1 = \begin{bmatrix}\beta_1\\ 0\end{bmatrix},\ B_2 = \begin{bmatrix}0\\ \beta_2\end{bmatrix}, \tag{2}$$

and z_1 and z_2 are the tank level heights in centimeters, u_1 and u_2 are the input voltages to the pumps, in decivolts. The time units are in minutes. The numerical values of the constants are: $r_{12} = 4.2$, $r_2 = 18$, $\beta_1 = 0.066$, $\beta_2 = 0.063$. We assume that the linearised model is valid for all the state and input variations under consideration.

This system is differentially flat (see [6], [7]) with z_1 and z_2 as the flat outputs. The inputs can be described in terms of these flat outputs and their derivatives. Overall, we obtain the following invertible-matrix relationship

$$\begin{bmatrix}u_1\\ u_2\\ z_1\\ z_2\end{bmatrix} = \begin{bmatrix}\frac{1}{r_{12}\beta_1} & \frac{1}{\beta_1} & -\frac{1}{r_{12}\beta_1} & 0\\ -\frac{1}{r_{12}\beta_2} & 0 & \frac{1}{r_{12}\beta_2}+\frac{1}{r_2\beta_2} & \frac{1}{\beta_2}\\ 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}z_1\\ \dot{z}_1\\ z_2\\ \dot{z}_2\end{bmatrix}. \tag{3}$$

The initial equilibrium point is

$$u_1^0 = 30,\quad u_2^0 = 20,\quad z_1^0 = 60,\quad z_2^0 = 50. \tag{4}$$

Hence we have

$$\begin{aligned}\bar{u}_1 &= u_1 + u_1^0, & \bar{u}_2 &= u_2 + u_2^0\\ \bar{z}_1 &= z_1 + z_1^0, & \bar{z}_2 &= z_2 + z_2^0\end{aligned} \tag{5}$$

where $\bar{u}_1$ is the absolute (non-linearised) value of the first input (and similarly for the other signals).

The actual inputs are constrained as

$$0 \le \bar{u}_1 \le 80, \text{ and } 0 \le \bar{u}_2 \le 40, \tag{6}$$

so that, for the linearised model,

$$-30 \le u_1 \le 50, \text{ and } -20 \le u_2 \le 20. \tag{7}$$

B. One-input, two-outputs model

In this paper we also consider (for illustration purposes and without loss of generality) the possibility of a fault in the system consisting in the total outage of the second pump. In order to detect a fault and reconfigure the trajectories properly we derive here the relevant model consisting of the same system with a single pump on the first tank. This model can be obtained from the first model (1) by setting $\bar{u}_2 = 0$, that is, $u_2 = -u_2^0$ (we assume the linearised model is still valid in this new situation) as follows

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} = A\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}B_1 & B_2\end{bmatrix}\begin{bmatrix}u_1\\ -u_2^0\end{bmatrix}. \tag{8}$$

The loss of an actuator (i.e., u_2 being fixed at the value u_2=$-u_2^0$, since $\bar{u}_2 = 0$) has rendered the model (8) nonlinear. We then linearise model (8) to obtain:

$$\begin{bmatrix}\dot{w}_1\\ \dot{w}_2\end{bmatrix} = A\begin{bmatrix}w_1\\ w_2\end{bmatrix} + B_1 u_1, \tag{9}$$

where $\bar{z}_1 = w_1 + w_1^0$ and $\bar{z}_2 = w_2 + w_2^0$. The new equilibrium point is

$$\begin{bmatrix}w_1^0\\ w_2^0\end{bmatrix} = A^{-1}B_2\, u_2^0 + \begin{bmatrix}z_1^0\\ z_2^0\end{bmatrix} = \begin{bmatrix}37.32\\ 27.32\end{bmatrix}. \tag{10}$$

Note that the equilibrium point for pump 1 does not change (i.e., $u_1^0 = 30$) and

$$w_1 = z_1 + z_1^0 - w_1^0 \quad \text{and} \quad w_2 = z_2 + z_2^0 - w_2^0. \tag{11}$$

The system (9) is also differentially flat, with w_2 as the flat output. The tank 1 level, w_1, and the [only] input, u_1, can be described in terms of w_2 and its derivatives, and conversely, w_2, $\dot{w}_2$, and $\ddot{w}_2$ can be described in terms of the states and input. This can be written using the invertible-matrix relation

$$\begin{bmatrix}u_1\\ w_1\\ w_2\end{bmatrix} = \begin{bmatrix}\frac{1}{\beta_1 r_2} & \frac{2}{\beta_1}+\frac{r_{12}}{\beta_1 r_2} & \frac{1}{\beta_1}r_{12}\\ 1+\frac{r_{12}}{r_2} & r_{12} & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}w_2\\ \dot{w}_2\\ \ddot{w}_2\end{bmatrix}. \tag{12}$$

III. Minimum-time trajectory generation

A. Brief summary of the procedure

From the flatness notion, we have that, for a controllable LTI system, every state and input can be expressed linearly in terms of the flat outputs and their derivatives. Conversely, the flat outputs and their derivatives are a linear combination of the states and the inputs. For the double tank system under healthy operation, this is given by (3).

With the B-splines concept [9], [10], one can finitely parameterise a continuous-time signal $y(t)$ using a number of basis function and their corresponding multipliers (*control points*):

$$y(t) = \Lambda_d(t)\, P, \quad t \in [t_0, t_f], \tag{13}$$

where $\Lambda_d(t)$ is a basis functions matrix of degree d, and P is the control points vector. Using a simple transformation (a matrix multiplication), derivatives of the signal $y(t)$ can also be parameterised by the same basis functions and control points, and thus, for example, an expression such as

$$u(t) = a_0\, y(t) + a_1\, \dot{y}(t) + a_2\, \ddot{y}(t), \quad t \in [t_0, t_f] \tag{14}$$

can be written as

$$u(t) = \Lambda_d(t)\mathcal{U}\, P, \quad t \in [t_0, t_f] \tag{15}$$

where $\mathcal{U}$ is a combined transformation matrix, given in this case by:

$$\mathcal{U} = a_0\, I + a_1 L_{d,d-1} M_{d,d-1} + a_2 L_{d,d-2} M_{d,d-2}, \tag{16}$$

where $L_{d,d-1}M_{d,d-1}$ and $L_{d,d-2}M_{d,d-2}$ are matrices that translate control points of a signal to the signal's first and second derivative spaces, respectively, see [2], [3] for a more detailed explanation. (Expression (14)—and equivalently (15)— corresponds, for example, to the first row of equation (12) with $u = u_1$, $y = w_2$, $a_0 = \frac{1}{\beta_1 r_2}$, $a_1 = \frac{2}{\beta_1} + \frac{r_{12}}{\beta_1 r_2}$ and $a_2 = \frac{r_{12}}{\beta_1}$; note that any other signal in (3) or in (12) can be treated in a similar way, including the case—as in (3)—with multiple flat outputs.). A useful feature of this parameterisation is that it results from properties of B-splines that the signal $u(t)$ is confined to be in the convex hull of $\mathcal{U}P$. This means that constraining the control points implies constraining the whole signal in the interval $[t_0, t_f]$. In other words, every signal and constraint can be mapped to the control points space of the B-splines parameterisation. This way, the problem of constrained trajectory generation can be cast into a standard quadratic programming problem in the splines control-points P-space (see [2], [3]).

In our approach to minimum-time control, it can be shown that the constraints form a polytope in the P-space whose shape changes as the end-time of the parameterisation is varied. This fact is exploited to iteratively search for a polytope that is minimally feasible, at which point a minimum time is reached. See [4] for a detailed exposition of the method, and [11] for an application to a real experimental plant.

B. *Application to the double-tank system*

Consider the healthy system (1). The tanks' levels are required to change, from the current equilibrium point to another equilibrium point (that is, rest-to-rest) as fast as possible; from $\bar{z}_1(t_0) = 60$, $\bar{z}_2(t_0) = 50$ to $\bar{z}_1(t_f) = 68$, $\bar{z}_2(t_f) = 60$ with $t_f - t_0$ as small as possible. The trajectory requirement for the flat outputs is then

$$\begin{aligned} &z_1(t_0) = 0, \quad \dot{z}_1(t_0) = 0, \quad z_2(t_0) = 0, \quad \dot{z}_2(t_0) = 0 \\ &z_1(t_f) = 8, \quad \dot{z}_1(t_f) = 0, \quad z_2(t_f) = 10, \quad \dot{z}_2(t_f) = 0. \end{aligned} \tag{17}$$

(Note that, from (3), the requirement of rest-to-rest trajectories imposes the first derivatives of both flat outputs to be zero at the end-points.) To compute the minimum-time trajectory for this system, using the procedure outlined in Subsection III-A we used Matlab with the cvx optimisation toolbox [12], and B-splines of degree 4, 39 control points for each flat output, and 17 iterations. The result is depicted in Fig. 2. Note that most of the time the input signals hit the constraints, which is a characteristic of time-optimal control (bang-bang). At the end of the trajectories, the inputs reach the new equilibrium values. The final time is $t_f = 7.1785$ mins.

IV. Algebraic Estimation and Residual-Based Fault Detection

In the proposed algebraic estimation method, we want to extract the control points that best fit a given measured signal. In turn, we can then utilise these control points to obtain all the other signals using the flatness parameterisation.

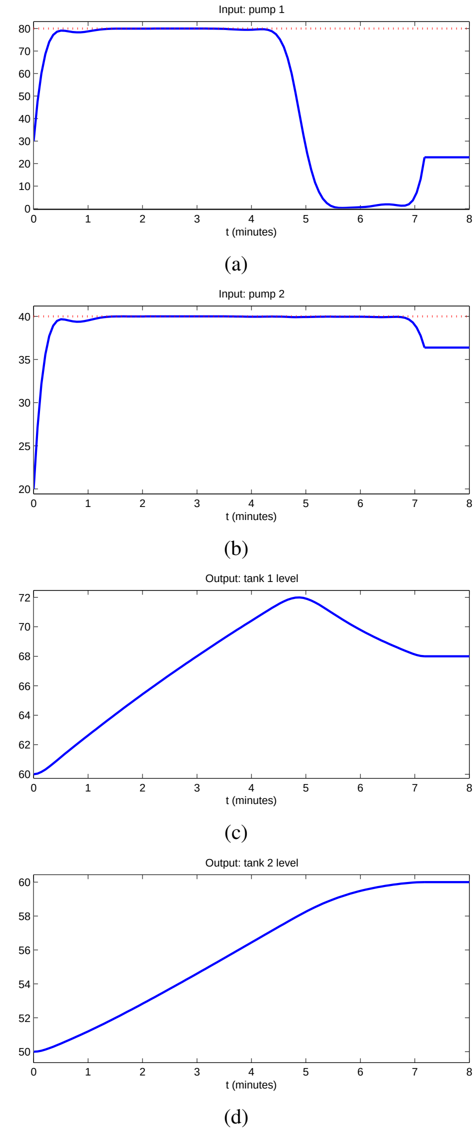


Fig. 2. Rest-to-rest trajectory. It can be seen that the input signals hit the constraints most of the time, which is characteristic of bang-bang control. The final time is $t_f = 7.1785$ mins.

A. *B-spline regression*

Common in statistics, this procedure will produce a trend curve of a given set of time-series data, or, in our case, samples of a signal corrupted with noise. This can be done by projecting the signal onto the column space of the basis functions.

For $\Lambda_d(t)$ defined over $t \in [t_0, t_f]$, define $\hat{\Lambda}_d$ as its sampled version, sampled at least $d+1$ times. Denote the sampling instant $k_i, i = 0, \ldots, M$, and $k_0 = t_0, k_M = t_f$. Now, given a set of time-series data (or noise-corrupted signal) $y(t)$, we construct the trend curve (or the smoothed signal) by using least-squares:

$$\bar{y} \triangleq \hat{\Lambda}_d (\hat{\Lambda}_d)^{+} \hat{y} = \hat{\Lambda}_d (\hat{\Lambda}_d^{\mathrm{T}} \hat{\Lambda}_d)^{-1} \hat{\Lambda}_d^{\mathrm{T}} \hat{y} \triangleq \hat{\Lambda}_d \hat{P}, \tag{18}$$

where $\hat{y} = [y(k_0) \ldots y(k_M)]^{\mathrm{T}}$ (in real applications the sampling time is usually uniform, so $k_{i+1} - k_i = \Delta$). Figure 3

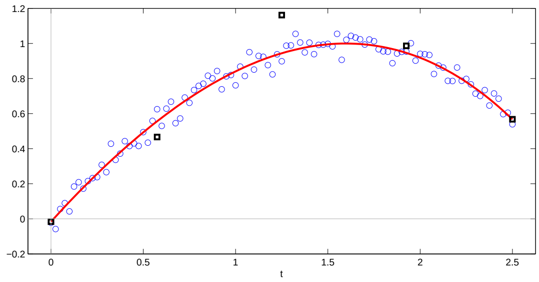


Fig. 3. An example of regression/noise filtering with B-Splines. The blue circles are the data to be regressed. The red solid lines are the B-Spline trend curve. The black squares are the control points corresponding to each basis function.

shows an example of the B-Spline regression described here. Hence we have that

$$\hat{P} \triangleq (\hat{\Lambda}_d)^{+}\hat{y} \tag{19}$$

defines the control points in the projected space. Once these estimated control points are obtained, they can be used to estimate other continuous-time signals. For example, we can readily have the following derivative estimation

$$\dot{\hat{y}}(t) = \Lambda_d(t) L_{d,d-1} M_{d,d-1} \hat{P}. \tag{20}$$

where $L_{d,d-1}M_{d,d-1}$ is a matrix that translates control points of a signal to the signal's derivative space. (See [2], [3] for the complete details).

B. Real-time noise filtering and derivative estimation

Here we extend the idea of B-Spline regression to real-time signal processing. The idea is similar to that of the previous subsection; however, since the signal is only available for past and current time instants, the technique described above cannot be applied in real time.

The technique we propose then is to use a sliding-window approach. The portion of the signal currently in the window is then regressed as explained above. The last sample is taken as the value of the filtered signal. The derivative signal, or any other signal, is then computed similarly. Figure 4 shows an example.

Note that if the signal being processed is the system's flat output, then using, for example, Eq. (3), one can obtain every other signal in one step. This procedure to obtain another signal in a flat linear system given the flat outputs, can be shown to be a process of FIR filtering.

C. Application to the double-tank system

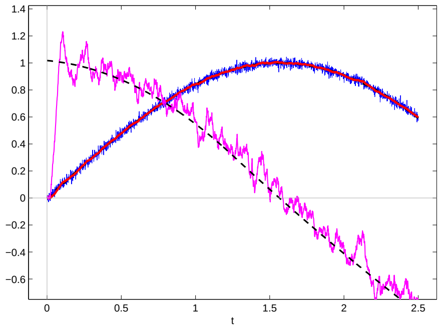
Using the derivative estimation method described above, we compute the following estimation of the inputs (see (3))

$$\hat{u}_1 \triangleq \frac{1}{r_{12}\beta_1} z_1 - \frac{1}{r_{12}\beta_1} z_2 + \frac{1}{\beta_1}\hat{\dot{z}}_1 \tag{21}$$

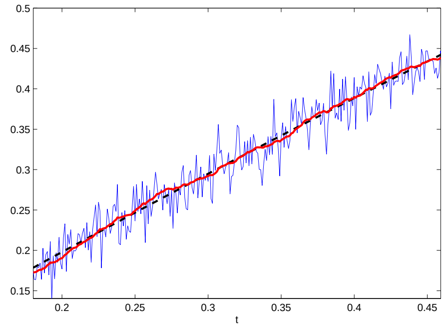
$$\hat{u}_2 \triangleq -\frac{1}{r_{12}\beta_2} z_1 + \frac{1}{\beta_2}\left(\frac{1}{r_{12}} + \frac{1}{r_2}\right) z_2 + \frac{1}{\beta_2}\hat{\dot{z}}_2. \tag{22}$$

Using the above estimations and the measured voltages sent to the pumps we define the residuals

$$R_1 \triangleq u_1 - \hat{u}_1 \tag{23}$$



(a)



(b)

Fig. 4. (a) An example of noise filtering and derivative estimation with B-Splines. (b) Zoom. The dashed "noiseless" lines are obtained from full-window B-Spline regression (hence non-causal). The red line and purple line are, respectively, the filtered signal and its estimated derivative, all using a sliding-window scheme. Here we used B-Splines of order one with two control points, with 0.1 seconds sliding window. The sampling time is 0.001 second.

$$R_2 \triangleq u_2 - \hat{u}_2. \tag{24}$$

We also use the single-input model (see (12)) to obtain

$$\hat{w}_1 \triangleq \left(1 + \frac{r_{12}}{r_2}\right) w_2 + r_{12}\hat{\dot{w}}_2, \tag{25}$$

that is, the estimation of the first tank's level from the second tank's measured level w_2 as seen from model (9). From this, we define the residual

$$R_3 \triangleq w_1 - \hat{w}_1. \tag{26}$$

Remark 1:

Under the total loss of the second actuator (and no other fault), it is expected that

1) $\hat{u}_1$ will correctly estimate u_1 and hence R_1 will stay unaffected.
2) $\hat{u}_2$ will estimate the real value of the second input, which will differ from the voltage sent to the second pump, and hence R_2 will be affected.
3) $\hat{w}_1$ will now correctly estimate w_1 (the first tank level from the point of view of model (9)), and hence R_3 will be zero. Note that this only happens if the (effective 'non-linearised' value of the) second input is zero. Thus, if the intended input to the second pump is non-zero, having R_3 zero indicates total loss of the second pump. This can be thought of as "model matching" of the plant under total loss of the second actuator.

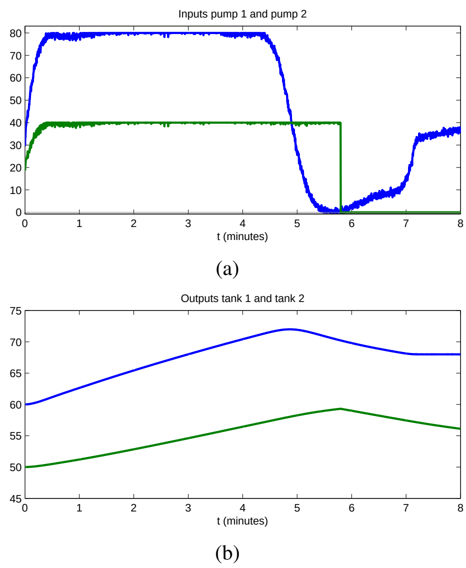


Fig. 5. (a) Inputs and (b) Outputs. The second pump fails at 5.8 mins.

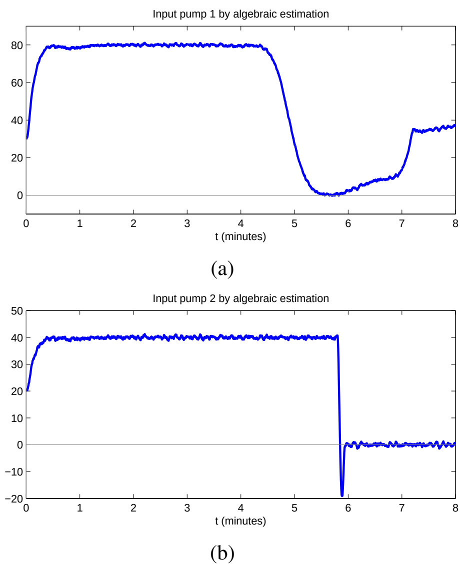


Fig. 6. Estimation of (a) the first input (from (21)) and (b) the second input (from (22)) corresponding to Fig. 5.

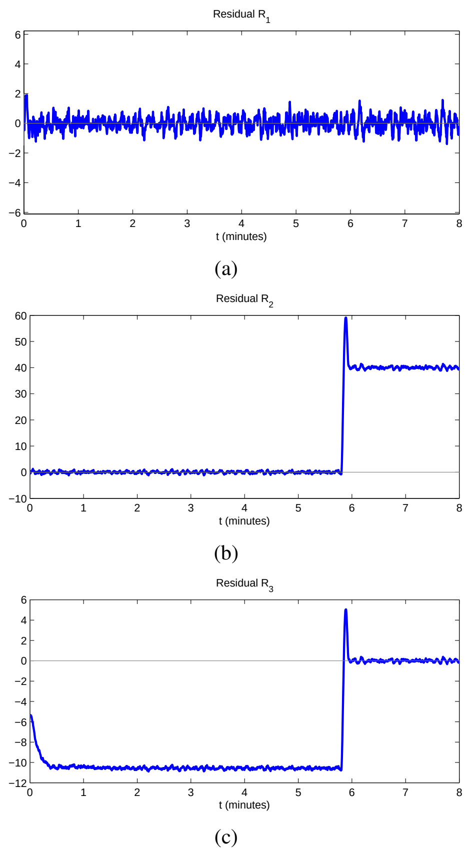


Fig. 7. Residuals R_1, R_2 and R_3 corresponding to Fig. 5.

The trajectories in Fig. 2 are now implemented in a simulation plant with a state feedback controller. For the derivative estimation, the signals are sampled every 0.2 seconds, the filter window is of 40 samples (= 8 seconds), and B-splines of degree 2 are used. In order to simulate a realistic situation, noise of magnitude 0.0017 was added to the output sensors of model (1) and model (9).

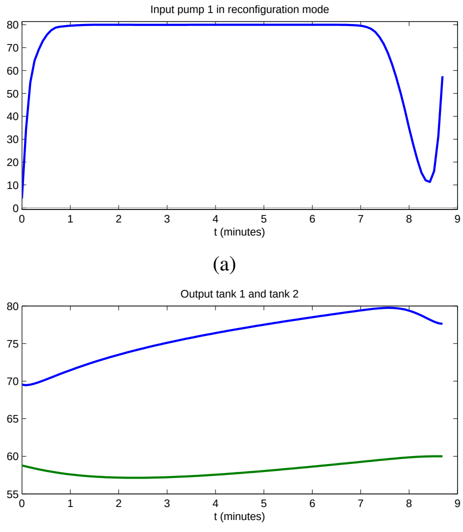
In the simulated scenario, the second pump is lost at 5.8 mins. The situation for the inputs and outputs is depicted in Fig. 5. Figure 6 shows the estimated value $\hat{\bar{u}}_1 = \hat{u}_1 + u_1^0$ and $\hat{\bar{u}}_2 = \hat{u}_2 + u_2^0$. It can be seen that the estimated values reflect the true inputs. Hence, the residuals behave as expected (see Remark 1 above), as depicted in Fig. 7.

Note that in a very similar way one can define other residuals that will indicate other faults in different system components. Due to space limitations, we have restricted ourselves to only illustrate the case of total outage of pump number 2.

V. Trajectory Reconfiguration

We apply the following thresholds for the residuals: for R_1 and R_2, if the signal's mean (moving average) is beyond ± 1.0, it is considered "nonzero"; for R_3, if the signal's mean (moving average) is within ± 0.5, then it is considered "zero". Using these thresholds, the loss of actuator 2 is detected and isolated 40 samples (or 8 seconds, the filter window) after the fault occurrence. The reconfiguration algorithm is required to recompute the new minimum-time trajectory in 12 seconds. This is done from the perspective of the one-input model (9). From Fig. 5(b) and 6(a), the initial states and input for reconfiguration are $w_1 = 32.21$, $w_2 = 31.46$, $u_1 = -26$. Since we lose one degree of freedom, we can only steer one output to a desired new target equilibrium value (the other output's target equilibrium value cannot be independently chosen). The objective is then chosen so as to keep the second tank's level original target equilibrium value of $\bar{z}_2 = 60$, or $w_2 = 32.68$. Using (12), all the requirements can be translated into the [single] flat output w_2 and its derivatives:

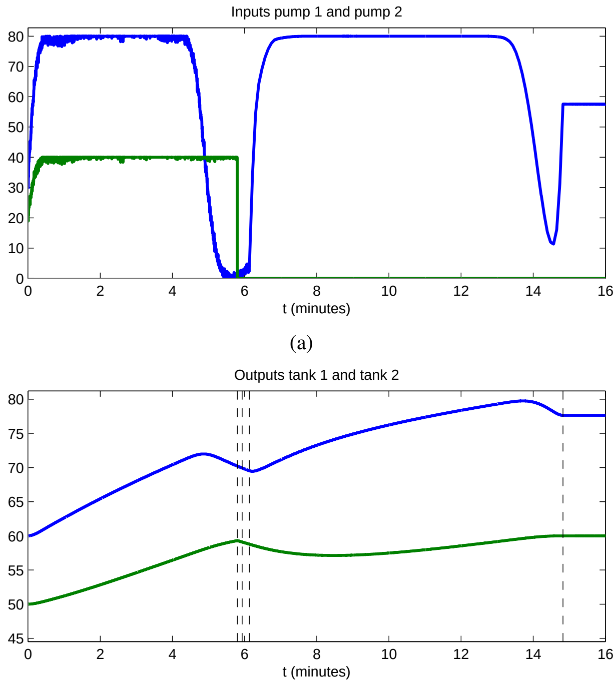
$$\begin{aligned} w_2(t_0) &= 31.46, & \dot{w}_2(t_0) &= -1.57, & \ddot{w}_2(t_0) &= 0.0097, \\ w_2(t_f) &= 32.68, & \dot{w}_2(t_f) &= 0, & \ddot{w}_2(t_f) &= 0, \end{aligned} \tag{27}$$



(a)

(b)

Fig. 8. Reconfigured trajectory. The final end-time is $t_f = 8.69$ minutes. See Fig. 9 for the overall trajectory.



(a)

(b)

Fig. 9. Overall trajectory. Corresponding to the dashed vertical lines in (b), from left to right: at 5.8 minutes the second pump is lost. Eight seconds later the fault is isolated. Twelve seconds later the remedy trajectory is applied. At $t_f = 14.82$ minutes, the new equilibrium point is reached.

where the values of $\dot{w}_2(t_0)$ and $\ddot{w}_2(t_0)$ are obtained from the measured $u(t_0)$, $w_1(t_0)$ and $w_2(t_0)$ by inverting the matrix in (12). Applying the procedure to compute the minimum-time trajectory under the new circumstances, the reconfigured trajectories are depicted in Fig. 8. The final end-time is $t_f = 8.69$ minutes. It can be seen that, to achieve the new equilibrium point, pump 1 has to go from a small value (due to its value at the particular moment the fault occurred in the initial trajectory) to the maximum value before it switches back and finally it reaches the target equilibrium value. Again, this is a typical bang-bang control solution.

The overall scenario can be seen in Fig. 9. In these figures, at 5.8 minutes the second pump is lost. At 5.9333 minutes, the fault is isolated. At 6.133 minutes, the remedy trajectory is executed. At 14.82 minutes the new equilibrium point is reached. These four time instants are indicated in Fig. 9(b) with vertical dashed lines.

VI. Conclusions

In this paper we have provided a unified treatment to the problems of constrained minimum-time trajectory generation, fault detection and identification, and trajectory reconfiguration. The integrated scheme that was presented allows to solve the three problems—traditionally dealt with separately—in a unified manner, using the same mathematical/computational tools; namely, differential flatness and B-splines. This, not only offers an elegant solution, but also has the potential to simplify the coding of the algorithms for the real-time application of the strategy. A case-study consisting of an input-constrained double-tank system has been analysed throughout the paper in order to illustrate the techniques in an intuitive manner.

References

[1] R. Bellman, I. Glicksberg, and O. Gross, "On the 'bang-bang' control problem," *Quarterly of Applied Mathematics*, vol. 14, pp. 11–18, 1956.

[2] F. Suryawan, J. De Doná, and M. Seron, "Splines and polynomial tools for flatness-based constrained motion planning," *International Journal of Systems Science*, 2011. `http://www.informaworld.com/10.1080/00207721.2010.549592`.

[3] F. Suryawan, J. De Doná, and M. Seron, "On splines and polynomial tools for constrained motion planning," in *Proc. 18th IEEE Mediterranean Conf. Control and Automation*, (Marrakech, Morocco), pp. 939–944, 23–25 June 2010.

[4] F. Suryawan, J. De Doná, and M. Seron, "Flatness-based minimum-time trajectory generation for constrained linear systems using B-splines," in *Proc. 18th IFAC World Congress*, (Milan, Italy), Aug 29–Sep 2, 2011. Available online at www.cdsc.org.au/Publications/pdf/SDS2011IFAC.pdf.

[5] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, June 1995.

[6] H. Sira-Ramírez and S. K. Agrawal, *Differentially Flat Systems*. New York: Marcel Dekker, 2004.

[7] J. Lévine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach*. Springer, 2009.

[8] F. Suryawan, J. De Doná, and M. Seron, "Fault detection, isolation, and recovery using spline tools and differential flatness with application to a magnetic levitation system," in *Proc. Conf. Control and Fault Tolerant Systems*, (Nice, France), pp. 293–298, Oct. 2010.

[9] C. de Boor, *A Practical Guide to Splines*. Springer, 1978.

[10] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 2nd ed., 1997.

[11] F. Suryawan, J. De Doná, and M. Seron, "Constrained minimum-time trajectory generation for a laboratory-scale magnetic-levitation system," *Submitted to 1st Australian Control Conference*, (Melbourne, Australia), November 2011. Available online at www.cdsc.org.au/Publications/pdf/SDS2011AuCC1.pdf.

[12] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," in *Global Optimization: from Theory to Implementation, Nonconvex Optimization and Its Applications* (L. Liberti and N. Maculan, eds.), pp. 155–210, New York: Springer, 2006. Available at `www.stanford.edu/~boyd/disc_cvx_prog.html`.