Measurement Feedback Disturbance Decoupling in Discrete-Time Nonlinear Systems

Arvo Kaldmäe\textsuperscript{a}, Õlle Kotta\textsuperscript{a,1}, Alexey Shumsky\textsuperscript{b}, Alexey Zhirabok\textsuperscript{b}

\textsuperscript{a}Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia
\textsuperscript{b}Far Eastern Federal University, Sukhanov street 8, 690990 Vladivostok, Russia

Abstract

The paper studies the disturbance decoupling problem by the dynamic measurement feedback for discrete-time nonlinear control systems. To address the problem the algebraic approach, called the algebra of functions, is applied, which allows the system description also depend on non-differentiable functions. A necessary and sufficient condition is given in terms of controlled and \((h,f)\)-invariant functions. Also, an algorithms are derived, which find invariant functions and the required feedback. The algorithms are implemented in Mathematica software which is made available over the internet.

Key words: disturbance rejection, output feedback control, nonlinear control systems, discrete-time systems, algebraic approaches

1 Introduction

The dynamic measurement disturbance decoupling problem (DDDPM) for nonlinear systems has been addressed only in a few papers \[2,3,6,7,10\]. Except \[7\] all the other papers address the continuous-time case and the papers \[2,3,6\] provide the solvability conditions within differential geometric framework. In the earliest paper \[6\] the feedback considered is restricted to the so-called pure dynamic measurement feedback (i.e the feedback that satisfies the condition (3) below) whereas \[2\] and \[3\] consider the general case but provide either only necessary conditions \[2\] or make additional assumptions \[3\]. The paper \[10\] (and its extension to the discrete-time case \[7\]) suggests sufficient algorithm-based condition for single-input single-output system with single measurement, applying the results (in terms of differential 1-forms) on input-output linearization by dynamic output feedback \[4\]. Moreover, note that \[8\] addresses the case when the measured output is the same as the output-to-be-controlled. To resume, the problem is old, but up to now has no full solution.

In this paper we address the DDDPM for discrete-time nonlinear control systems and the problem statement is similar to that of \[6\]. Especially, note that the controller is designed to be a suitable subsystem of the original system and initial state of the compensator has to be chosen in accordance with that of the system. Such type of controller reduces the dimension of the closed-loop system compared for example with those in \[2,3,7,10\] and has contact points with the ‘regular interconnection’ as addressed in \[9\]. Note that in the solutions of \[2,3,7,10\] the dimension of the closed-loop system is the sum of those of the plant and the controller whereas in this paper (and in \[6\]) it is equal to the state of the plant.

The problem is studied within the algebraic approach, called the algebra of functions \[11\], and are related to lattice theory. Our choice is not based on the belief that the algebraic tools are superior than those of the differential geometry, but were rather determined by the fact that the extensions of the differential geometric tools for discrete-time systems are not as well developed and universally accepted as those for the continuous-time case \[5,7\]. Moreover, the tools of algebra of functions seem to be well-suited for the problem statement adopted in this paper, and the system description may also depend on non-smooth functions. In particular, we aim to get the necessary and sufficient solvability conditions together with the complete algorithmic solution. Comparison of our results with those from \[7\] and \[6\] are given in Sec-

---

Email addresses: arvo@cc.ioc.ee (Arvo Kaldmäe), kotta@cc.ioc.ee (Õlle Kotta), shumsky@mail.primorye.ru (Alexey Shumsky), zhirabok@mail.ru (Alexey Zhirabok).

\textsuperscript{1} Tel.: +372 6204153; fax: +371 6204151

Preprint submitted to Automatica 23 May 2013
tion 4. Since our tools are not well-known, the webMathematica based software has been developed so that anyone can use with only an internet browser [1].

2 Problem statement

Consider a discrete-time nonlinear control system

\[ \begin{align*}
x(k + 1) &= f(x(k), u(k), w(k)), \\
y(k) &= h(x(k)), \quad y_s(k) = h_s(x(k)),
\end{align*} \tag{1} \]

where \( x \in X \subseteq \mathbb{R}^n \) is the state, \( u \in U \subseteq \mathbb{R}^m \) is the control, \( w \in W \subseteq \mathbb{R}^p \) is the unmeasurable disturbance, \( y \in Y \subseteq \mathbb{R}^j \) is the measured output and \( y_s \in Y_s \subseteq \mathbb{R}^l \) is the output-to-be-controlled. The disturbance deconvolving problem under a dynamic measurement feedback (DDDPM) can be stated as follows: find a dynamic measurement feedback of the form

\[ \begin{align*}
z(k + 1) &= F(z(k), y(k), v(k)) \\
u(k) &= G(z(k), y(k), v(k)),
\end{align*} \tag{2} \]

where \( v \in V \subseteq \mathbb{R}^m \) and

\[ \text{rank}[\partial G/\partial v] = m, \tag{3} \]

such that the values of the outputs-to-be-controlled \( y_s(k) \), for \( k \geq 0 \), of the closed-loop system are independent of the disturbances \( w \). Note that we call the compensator, described by (2) regular, since it generally defines the \((y,z)\)-dependent one-to-one correspondence between the variables \( v \) and \( u \). One says that the disturbance deconvolving problem is solvable via static output feedback if \( u(k) = G(y(k), v(k)) \).

3 The algebra of functions

To address the DDDPM, the mathematical approach called the algebra of functions [11] will be used. We recall briefly the definitions and concepts to be used in this paper. The elements of algebra of functions are vector functions and its main ingredients are: (1) relation of partial preorder, denoted by \( \leq \), (2) binary operations, denoted by \( \times \) and \( \oplus \), (3) binary relation, denoted by \( \Delta \), (4) operators \( m \) and \( M \). The first two elements are defined on the set \( S_X \) of vector functions with the domain being the arbitrary set \( S \) whereas the last two are defined for the set \( S_X \) of vector functions with the domain being the state space \( X \).

Definition 1 (Relation of partial preorder) Given \( \alpha, \beta \in S_X \), one says that \( \alpha \leq \beta \) iff there exists a function \( \gamma \) such that \( \beta(s) = \gamma(\alpha(s)) \) for \( \forall s \in S \).

Definition 2 (Equivalence) If \( \alpha \leq \beta \) and \( \beta \leq \alpha \), then \( \alpha \) and \( \beta \) are called strictly equivalent, denoted by \( \alpha \equiv \beta \).

Note that the relation \( \equiv \) is reflexive, symmetric and transitive. The equivalence relation divides the set \( S_X \) into the equivalence classes containing the equivalent functions. If \( S_X \equiv \) is the set of all these equivalence classes, then the relation \( \leq \) is partial order on this set. Recall that a lattice is a set with a partial order where every two elements \( \alpha \) and \( \beta \) have a unique supremum (least upper bound) \( \sup(\alpha, \beta) \) and an infimum (greatest lower bound) \( \inf(\alpha, \beta) \). The equivalent definition of the lattice as an algebraic structure with two binary operations \( \times \) and \( \oplus \) may be given if for every two elements both operations are commutative and associative and moreover, \( \alpha \times (\alpha \oplus \beta) = \alpha, \alpha \oplus (\alpha \times \beta) = \alpha \). The equivalence follows from the definition the binary operations \( \times \) and \( \oplus \) as

\[ \alpha \times \beta = \inf(\alpha, \beta), \quad \alpha \oplus \beta = \sup(\alpha, \beta). \tag{4} \]

Therefore, the triple \((S_X \equiv, \times, \oplus)\) is a lattice. In lattice theory it is customary not to operate with \( \inf(\alpha, \beta) \) and \( \sup(\alpha, \beta) \) but with binary operations \( \times \) and \( \oplus \), respectively. In the simple cases, (4) may be used to compute \( \alpha \oplus \beta \). The rule for operation \( \times \) is simple: \((\alpha \times \beta)(s) = [\alpha(s), \beta(s)]^T\). However, the product may contain functionally dependent components that have to be found and removed. Note that there exist two special vector functions \( 0 \) and \( 1 \), such that for every function \( \alpha \), \( 0 \leq \alpha \leq 1 \).

Example 3 (Computation of the functions \( \alpha \times \beta \) and \( \alpha \oplus \beta \)). Let \( S_X = \mathbb{R}^3 \), \( \alpha(s) = [s_1 + s_2, s_3]^T \), \( \beta(s) = [s_1 s_3, s_2 s_3]^T \). To compute \( \alpha \times \beta \), remove the functionally dependent component \( s_2 s_3 \) in \([\alpha(s), \beta(s)]^T = [s_1 + s_2, s_3, s_1 s_3, s_2 s_3]^T\) to get \((\alpha \times \beta)(s) = [s_1 + s_2, s_3, s_1 s_3, s_2 s_3]^T\). Clearly, by Definition 1, \( \alpha \times \beta \leq \alpha \) and \( \alpha \times \beta \leq \beta \) since \( \alpha_1 = (\alpha \times \beta)_1, \alpha_2 = (\alpha \times \beta)_2, \beta_1 = (\alpha \times \beta)_3, \beta_2 = (\alpha \times \beta)_1(\alpha \times \beta)_2 - (\alpha \times \beta)_3 \), and therefore both \( \alpha \) and \( \beta \) can be expressed via components of \( \alpha \times \beta \). Moreover, by Definition 1, \( \alpha \leq s_3(s_1 + s_2) \) and \( \beta \leq s_3(s_1 + s_2) \) and therefore, \((\alpha \oplus \beta)(s) \equiv s_3(s_1 + s_2)\).

Definition 4 (Binary relation \( \Delta \)) Given \( \alpha, \beta \in S_X \), there exists a function \( f_x \) such that for all \((x, u, w) \in X \times U \times W\), \((\alpha, \beta) \in \Delta \iff \beta(f(x, u, w)) = f_x(\alpha(x), u, w)\). When \((\alpha, \beta) \in \Delta\), it is said that \( \alpha \) and \( \beta \) form an ordered pair.

Binary relation \( \Delta \) is used for definition of the operators \( m \) and \( M \).

Definition 5 Operator \( m(\alpha) \) is a function in \( S_X \) that satisfies the following conditions \( (i) \) \((\alpha, m(\alpha)) \in \Delta\), \( (ii) \) if \((\alpha, \beta) \in \Delta\), then \( m(\alpha) \leq \beta \).

Definition 6 Operator \( M(\beta) \) is a function in \( S_X \) that satisfies the following conditions \( (i) \) \((M(\beta), \beta) \in \Delta\), \( (ii) \) if \((\alpha, \beta) \in \Delta\), then \( \alpha \leq M(\beta) \).

From Definitions 5 and 6 it is obvious that given \( \alpha \), \( m(\alpha) \) is the minimal function, forming a pair with \( \alpha \), and given
\( \beta, \textbf{M}(\beta) \) is the maximal function, forming a pair with \( \beta \).

**Computation of the operator \( \textbf{m} \).** It has proven that the function \( \gamma \) exists that satisfies the condition \((\alpha \times u) \circ f \equiv \gamma(f)\); define \( \textbf{m}(\alpha) \equiv \gamma, \) see [11]. The examples how to compute \( \gamma \) may be found in [11].

**Computation of the operator \( \textbf{M} \).** In the special case when \( \beta(f(x, u)) \) can be represented in the form \( \beta(f(x, u)) = \sum_{i=1}^{a} a_{i}(x) b_{i}(u) \) where \( a_{1}(x), a_{2}(x), \ldots, a_{d}(x) \) are arbitrary functions and \( b_{1}(u), b_{2}(u), \ldots, b_{d}(u) \) are linearly independent, then \( \textbf{M}(\beta) := a_{1} \times a_{2} \times \cdots \times a_{d} \). For the general case, see [11].

### 4 Problem solution

Find first a minimal (containing the maximal number of functionally independent components) vector function \( \alpha^{0}(x) \) such that its forward shift \( \alpha^{0}(f(x, u, w)) \) does not depend on the unmeasurable disturbance \( w \). Note that, the components of \( \alpha^{0}(x) \) are scalar functions with relative degree (with respect to the disturbance \( w \)) two or more. In the smooth case \( \alpha^{0} \) may be found as the solution of the PDE \( \partial^{2}/\partial u^{2}[\alpha^{0}(f(x, u, w))] \equiv 0 \). The function \( \alpha^{0}(x) \) plays a key role in Algorithms 1, 2 and 3 below, and though it is not unique, all possible choices are equivalent functions. Moreover, applying the operators \( \textbf{m} \) and \( \textbf{M} \) to equivalent functions will yield again equivalent functions. Therefore, the results of Algorithms 1, 2 and 3 will be the same for different choices of \( \alpha^{0}(x) \), up to the function equivalently.

**Definition 7** The vector function \( \alpha \) is said to be \((h, f)\)-invariant if \((\alpha \times h, \alpha) \in \Delta \). In case \( h = I \), function \( \alpha \) is said to be \( f \)-invariant.

**Definition 8** The vector function \( \alpha \) is said to be a controlled invariant if there exists a regular static state feedback \( u = G(x, v) \) such that function \( \alpha \) is \( f \)-invariant for the closed-loop system.

Definition 7 is a generalization for the discrete-time systems of the form (1) the concept of conditioned invariant (called alternatively \((h, f)\)-invariant) distribution (or codistribution) for continuous-time systems, as given for example in [6]. From one side, Definition 7 is a bit more general since the function \( f \) in (1) as well as a function \( \alpha \) are not necessarily smooth nor even differentiable. From the other side the definition in [6] is more general since the invariant distribution is not necessarily involutive (integrable). Definition 8 is a reformulation of a controlled invariant distribution, defined in [5].

The next Theorem gives a condition to check if a system is disturbance decoupled or not. The proof of this Theorem is obvious.

**Theorem 9** System (1) is disturbance decoupled if and only if there exists a \( f \)-invariant function \( \xi \) such that \( \alpha^{0} \leq \xi \leq h \).

To solve the DDDPM, we search for the state transition map \( F \) of the compensator (2) as a forward-shift of certain vector function \( \alpha(x) \). Therefore, the state \( z(k) \) of the compensator (2) is defined by \( z(k) = \alpha(x(k)) \). Then the dynamics \( F \) of compensator (2) is a part of the dynamics \( z(k + 1) := f(x(k), G(z(k), y(k), v(k)), w(k)) := f(x(k), v(k)) \). Since \( F \) is not allowed to depend on the disturbance \( w(k) \) and \( \alpha^{0} \) is the minimal vector function whose forward-shift does not depend on the disturbance \( w(k) \), the following condition must be satisfied: \( \alpha^{0} \leq \alpha \).

**Theorem 10** System (1) can be disturbance decoupled by feedback (2) where \( z(k) = \alpha(x(k)) \) if and only if the vector function \( \alpha \) (satisfying \( \alpha^{0} \leq \alpha \)) is \((h, f)\)-invariant and there exists a controlled invariant function \( \xi \) such that
\[
\alpha^{0} \leq \alpha \leq \xi \leq h.
\]

**PROOF.** Necessity. Assume that there exists a feedback (2) where \( z(k) = \alpha(x(k)) \) that solves the DDDPM. Then, by Theorem 9, there exists a \( f \)-invariant function \( \xi \) such that \( \alpha^{0} \leq \xi \leq h \). Since feedback (2) solves the DDDPM, \( \xi \) is \( f \)-invariant under the feedback \( u = G(x, v) = G(z, y, v) \) and thus the function \( \xi \) is controlled invariant. In (2), function \( F \) depend only on \( z = \alpha(x), y \) and \( v \), which means that \( \alpha(x) \) is clearly \((h, f)\)-invariant. Because \( \alpha \) is \((h, f)\)-invariant, then \( \alpha \times h \leq \textbf{M}(\alpha) \) and \( \alpha(x) \times h(x) \times u \leq \alpha(f(x, u)) \), i.e. the function \( \alpha(f(x, u)) \) can be expressed via \( \alpha(x), h(x) \) and \( u \). In the closed-loop system the control \( u \) is replaced with \( G(z, y, v) = G(\alpha(x), h(x), v) \) which can be expressed via \( \alpha(x), h(x) \) and \( v \) as well. Therefore, \( \alpha \) is \((h, f)\)-invariant. Since \( \xi \) is \( f \)-invariant, then \( \alpha \leq \xi \).

Sufficiency. Since function \( \xi \) is controlled invariant, there exists a static state feedback \( u = G(x, v) \) such that \( \xi(x(k + 1)) = \chi(\xi(x(k)), v(k)) \). Since \( \alpha^{0} \leq \xi \leq h \), then by Theorem 9, the closed-loop system is disturbance decoupled. It remains to show that function \( G \) depends only on variables \( z, y \) and \( v \). Since \( \alpha \leq \xi \), then \( \textbf{M}(\alpha) \leq \textbf{M}(\xi) \) and therefore from the definitions of \( (h, f)\)-invariant function and operator \( \textbf{M} \), one gets \( \alpha \times h \leq \textbf{M}(\xi) \). By definition of the operator \( \textbf{M} \), \( \textbf{M}(\xi)(\chi(x(k)), v(k)) \times u(k) \leq \xi(x(k + 1)) \). Thus \( \alpha \times h(x(k)) \times u(k) \leq \textbf{M}(\xi)(\chi(x(k)), v(k)) \times u(k) \leq \xi(x(k + 1)) = \chi(\xi(x(k)), v(k)) \). This means that \( \chi \) can be written in terms of \( z, y \) and \( v \) and then the function \( G \) depends also only on \( z, y \) and \( v \). ■

2 This is in accordance with [6], where the similar relation between the states of the control system and compensator is used to prove Theorem 3.7.
The next algorithm is used to compute the minimal \((h, f)\)-invariant vector function \(\alpha\), that satisfies the condition \(\alpha^0 \leq \alpha\).

**Algorithm 1**

**Given** \(\alpha^0\), compute recursively for \(i \geq 1\), using the formula \(\alpha^{i+1} = \alpha^i \ominus m(\alpha^i \times h)\), the sequence of non-decreasing functions \(\alpha^0 \leq \alpha^1 \leq \alpha^2 \leq \ldots \leq \alpha^i \leq \ldots\). By Theorem 1 in [8], there exists a finite \(\alpha^i \not\equiv \alpha^{i+1}\) but \(\alpha^i+1 \equiv \alpha^i\), for all \(i \geq 1\). Define \(\alpha := \alpha^i\).

Since \(\alpha\) is \((h, f)\)-invariant, i.e. \((\alpha \times h, \varphi) \in \Delta\), then by Definition 4 there exists a function \(F\) such that \(\alpha(f(x, u, w)) = F(\alpha(x), h(x), u)\). Because \(\alpha^0 \leq \alpha\), i.e. \(\alpha = \psi(\alpha_0)\) for some \(\psi\), then from the definition of \(\alpha^0\), \(\alpha(f(x, u, w))\), and therefore also \(F\), do not depend on \(w\).

Define the function \(z = \alpha(x) : X \rightarrow Z\), and construct the system

\[
z(k + 1) = \alpha(f(x(k), u(k), w(k))) = F(z(k), y(k), u(k)). \tag{6}
\]

Algorithm 2 below computes, if they exist, the controlled invariant function \(\xi\) that satisfies (5), and the feedback that solves the DDDPM. Before presenting the main algorithm of the paper we give a brief overview of it. The first step finds the \((h, f)\)-invariant function \(\alpha\), that satisfies \(\alpha^0 \leq \alpha \leq h\). It also defines the dynamics of the controller, up to the substitution of \(G(z, y, v)\) (found at Step 4) for \(u(t)\). The 4th step which relies on the solution of the set of nonlinear algebraic equations, finds the output equation of the controller. The intermediate Steps 2 and 3 construct the abovementioned set of nonlinear equations, analyzing the interdependent structure of the output and the state transition map of the controller dynamics \(F\). Step 5 allows to check whether the feedback constructed at the previous step to achieve disturbance decoupling for some output components may degrade the situation for the other components.

**Algorithm 2 Step 1.**

**Given** \(\alpha^0\), find, by Algorithm 1, the minimal \((h, f)\)-invariant function \(\alpha\) satisfying the condition \(\alpha^0 \leq \alpha\). If \(\alpha \leq h\) is not valid, then stop since the DDDPM is not solvable by Theorem 10. Otherwise, for \(\alpha \leq h\), is not valid and let \(\alpha = \psi(\alpha_0)\), where \(\alpha_0\) is the minimal \((h, f)\)-invariant function \(\alpha\) satisfying the condition \(\alpha_0 \leq \alpha\).

**Algorithm 2 Step 2.**

Split the vector \(y\) into two disjoint subvectors \(y_g\) and \(y_b\), some of them possibly empty:

1. \(y_i = h_i(x)\) is a component of \(y_g\) if either (a) the inequality \(\alpha \leq h_i\) holds, that is \(y_i = h_i(x)\) can be expressed in terms of \(z = \alpha(x)\), or (b) \(F\) does not depend on \(y_i\).

\(^3\) The subindices \(g\) and \(b\) stand for the words ‘good’ and ‘bad’.

(ii) the remaining \(y_i\)’s are the components of \(y_b\).

If \(y_b\) is an empty subvector then the system is already disturbance decoupled. Otherwise, for all \(y_i \in y_b\), find \(z_j\) such that \(F_j(z, y, u)\) depends on \(y_i\) and does not depend on \(u\). Denote by \(Z_b\) the set of all such \(z_j\)’s. If \(Z_b = \emptyset\), then go to Step 4.

**Step 3.**

Set \(Z_{vb} = \emptyset\). For each \(z_j \in Z_b\) find the function \(F_j(z, y, u)\) depending on \(z_j\). If \(F\) depends also on the control \(u\), then add \(z_j\) into \(y_b\), otherwise insert the respective \(z_j\) into \(Z_{vb}\). If \(Z_{vb} \cap Z_b \not= \emptyset\) then stop since the algorithm does not give a solution. Otherwise set \(Z_b = Z_{vb}\) and repeat Step 3 until \(Z_b\) remains unchanged or \(Z_{vb} = \emptyset\).

**Step 4.**

Find in \(F\) all terms of the form \(\gamma_i(z, y, u)\), \(i = 1, \ldots, r\), depending on \(y_i\) and \(u\). Let \(r\) be the number of such terms\(^4\). Assume that \(r \leq m^5\). Denote \(\gamma := [\gamma_1, \ldots, \gamma_r]^T\) and let rank \((\partial \gamma / \partial u)\) := \(q\) everywhere except perhaps on a set of measure zero. Denote \(u^1 = [u_1, \ldots, u^t_q]^T\) and \(u^2 = [u_{q+1}, \ldots, u_n]^T\) and split the vector \(v\) of new inputs in a similar manner. After a possible reordering the control components one may assume that rank \((\partial \gamma / \partial u^1)\) = \(q\). Note that\(^5\) the equation \(\gamma(z, y, u) = v^1\) can be solved (generically) uniquely for \(u^1\):

\[
u^1 = G^1(z, y, v^1, u^2). \tag{7}
\]

Furthermore, set \(u^2 = v^2 := G^2(z, y, v)\) and substitute in (7) \(v^2\) for \(u^2\). That way, we get

\[
u = G(z, y, v). \tag{8}
\]

**Step 5.**

Substitute in (6) \(G^1(z, y, v)\) for \(u^1\). Denote by \(f\) a vector function \(f\) for closed-loop system. Using Algorithm 3, find maximal \((h, f)\)-invariant function \(\xi = [\xi_1, \ldots, \xi_m]^T\) such that \(\xi \leq h\). If functions \(\xi_i(x(k + 1))\) depend on outputs \(y_i\) from \(y_b\) and also on control \(u\), then return to Step 4, otherwise return to Step 2. If functions \(\xi_i(x(k + 1))\) do not depend on outputs \(y_i\) from \(y_b\), then compensator (6), (8) solves the DDDPM.

The algorithm below is dual to Algorithm 1 and, besides being of interest itself, will be applied below to lower the dimension of the compensator (6), (8), see Remark 1.

**Algorithm 3 (Computation of the maximal \((h, f)\)-invariant function \(\delta\) satisfying the condition \(\delta \leq \delta^0\)).**

\(^4\) Note that not necessarily (though it may happen for some \(i\)) \(\gamma_i = F_i\)\(^5\).

\(^5\) In case \(r > m\), there exist several choices for functions \(\gamma_i\) to construct the system of equations \(\gamma_i(\cdot) = v_i, i = 1, \ldots, m\).

\(^6\) In general, the solvability of the equation is guaranteed by the Implicit Function Theorem that does not hold for non-smooth functions. Therefore, for Step 4 to be applicable \(\gamma\) is not allowed to contain \(u\) as an argument of non-smooth function.
**Step 1.** Set \( i := 0 \).

**Step 2.** Compute the function \( \gamma^i = M(\delta^i) \).

**Step 3.** If the components of the vector function \( \gamma^i \) can be expressed in terms of the components of the function \( h \times \delta^i \times \delta^i \times \cdots \times \delta^i \), then go to Step 5.

**Step 4.** Find the vector function \( \delta^{i+1} \) with minimal number of components, satisfying the condition \( h \times \delta^i \times \delta^i \times \cdots \times \delta^{i+1} \leq \gamma^i \). Set \( i := i + 1 \) and go to Step 2.

**Step 5.** Define \( \delta := \delta^0 \times \delta^1 \times \cdots \times \delta^i \).

**Remark 1** Some components of function \( G \) in (8) may be independent of the variable \( z \); the corresponding expressions give the static part of the measurement feedback. In order to reduce the dimension of the dynamic part of (2), collect all the components of \( z \) that show up on the right-hand side of (8), and denote them by \( z^0 = \rho(z) \). Find, by Algorithm 3, the maximal \((h, f)\)-invariant function \( \tilde{\rho} \) satisfying the condition \( \tilde{\rho} \leq \rho(x) \) (this function always exists: in the worst case, \( \tilde{\rho} \equiv \alpha \)). Denote \( z_* = \tilde{\rho}(x) \) and construct the system

\[
z_*(k+1) := \tilde{\rho}(f(x(k), u(k), w(k))) := F_1(z_*(k), y(k), u(k)).
\]

This system defines the compensator of the minimal dimension.

It is not an easy task to compare the results of this paper with earlier results even if to focus only on analytic systems. Though this paper and [7] assume basically the same structure (2) of the compensator there exist a few differences. In this paper the state \( z \) of the compensator is a function of the measurement, and the dimension of the closed-loop system equals the dimension \( n \) of the original system. However, in [7] the state of the compensator is a function of the measurement \( y \), the control \( u \), and their forward shifts. The problem statement in [7] is a bit more general since it does not assume that in (2) \( \text{rank}[\partial G/\partial v] = m \), and regularity of the compensator is defined by its invertibility (in the sense of Singh algorithm). From the other side the dimension of the closed-loop system equals \( n + q \). Because of different problem statements our results have to be considered as complimentary to those of [7]: there are examples when the results of [7] do not yield a solution and our’s do, but there are the other examples when the opposite is true.

In spite of using different tools, certain geometric constructions in [6], i.e. distributions and relations between them, have clear counterparts in terms of vector functions and their (partial) preorder. We will comment these connections below. For smooth systems, the distribution \( \mathcal{P} = \text{span}\{p(x)\} \) in [6] corresponds to the vector function \( \alpha^0 \) in the sense that the annihilator of the involutive closure of \( \mathcal{P} \), \( \mathcal{P}^\perp = \text{span}\{\alpha^0\} \). In the similar manner, the distributions \( \Delta^1 \) and \( \Delta^2 \) in [6] correspond to the controlled invariant function \( \xi \) and the \((h, f)\)-invariant function \( \alpha \), respectively. Note that since we work with functions, the distributions that correspond to them are always involutive. Then conditions of Theorem 10 are in complete agreement with conditions in Theorem 3.6 of [6].

## 5 Discussion and examples

The Mathematica-based symbolic software has been developed that is made available over the internet using webMathematica tools for researchers having no access to Mathematica. The developed website is available at [1], and allows to handle the main operations and operators of the algebra of functions, like \( \times \) and \( \oplus \), \( \mathfrak{m} \) and \( M \). Moreover, the Algorithms 1-2 are implemented. The examples below may be run on this website, the website also lists a number of additional examples. To simplify the exposition, we use below the symbols \( x^+ \) and \( x \) to denote \( x(k+1) \) and \( x(k) \), respectively, and use the similar notations for the other variables.

**Example 11** The system in Figure 1 is a typical sub-system in many applications and consists of linear sub-systems \( W_1 = k_1/(1 + T_1 \sigma) \), \( W_2 = k_2/(1 + T_2 \sigma) \), \( W_3 = k_3 T_3 \sigma/(1 + T_3 \sigma) \), \( W_4 = k_4 \sigma \) and saturation operation,

\[
\sigma(z) = \begin{cases} 
z, & \text{if } |z| \leq z_0 \\
z_0 \text{sign } z, & \text{if } |z| > z_0 
\end{cases}
\]

that corresponds to the amplifier. Here \( k_1 \div k_5 \), are real coefficients, \( k \) is a discretization step, \( T_1, T_2 \) are certain time constants and \( T_3 \) may be considered as unknown function of disturbance because of the unexpected changes in the feedback loop.

**Figure 1.**

The Euler discretization of this system is described by the equations:

\[
x_1^+ = k k_4 x_2 + x_1 \\
x_2^+ = \frac{k k_3}{T_2} \sigma(x_3) + x_2(1 - \frac{k}{T_2})
\]
\[ x^+_1 = \frac{k}{T_1} (k_1 k_3 (u - x_1) - k_1 k_3 (x_2 - x_4)) + x_3 (1 - \frac{k}{T_1}) \]
\[ x^+_4 = \frac{k}{T_3(w)} x_2 + x_4 (1 - \frac{k}{T_3(w)}) \]
\[ y = k_3 (x_2 - x_4), \quad y_s = x_1. \]

Find a minimal vector function \( \alpha'(x) \) such that its forward shift does not depend on \( T_3(w) \): \( \alpha'(x) = [x_1, x_2, x_3]^T \). Because \( \alpha'^0 \times h = \text{id}_X \) and \( m(\text{id}_X) = \text{id}_X \), then \( \alpha'^0 \oplus m(h \times \alpha'^0) = \alpha'^0 \) and \( \alpha^1 = \alpha'^0 \). Therefore \( \alpha(x) = \alpha'^0(x) = [x_1, x_2, x_3]^T \). According to Step 1, one has \( Z_1 = \{z_1\} \) and
\[
\begin{align*}
\begin{pmatrix}
kkz + z + 1 \\
k(k_5 (u - z_1) - k_1 y) + z_3 (1 - \frac{k}{T_1})
\end{pmatrix}
\end{align*}
\]

where \( z(0) = [x_1(0), x_2(0), x_3(0)]^T \). It is easy to check that \( y = y = \emptyset, \quad Z_0 = \emptyset \) because \( F_0 \) depends on \( u \) and \( y \). Obviously (see Step 4), one can set \( v = k_1 k_5 (u - z_1) - k_1 y, \quad r = 1 \) and so \( u = (v + k_1 y)/k_1 k_5 + z_1 \). Substituting \( u \) into (9) we get that the maximal \( (h, f) \)-invariant function \( \xi \) such that \( x \leq h \) is \( \xi = [x_1, x_2, x_3]^T \). Because \( F \) does not depend on \( y \) in \( f \), function \( \xi \) is \( f \)-invariant and condition (5) is satisfied: \( \alpha^0 = \alpha = \xi = [x_1, x_2, x_3]^T \leq h_0 = x_1 \). Then \( u = (v + k_1 y)/k_1 k_5 + z_1 \) with (9) gives the solution to the DDDPM.

**Example 12** Consider the Euler discretization of the example from [3], where for simplicity we have chosen \( a_1(x_1) = \text{sign}(x_1 + 1), \quad b_1(x_1) = x_1, \quad \varphi(x_4) = x_4, \quad a_2(x_1, x_3) = x_1 x_3, \quad b_2(x_1) = x_1: \)
\[
\begin{align*}
x^+_1 &= x_1 x_2 + u + x_1, \quad x^+_2 = \text{sign}(x_1 + 1) + x_4 \\
x^+_3 &= x_1 x_3 + x_1 x_2 + x_3 + u, \quad x^+_4 = x_3 - x_4 \\
y_1 &= x_1, \quad y_2 = x_4, \quad y_s = x_1.
\end{align*}
\]

Note that \( \alpha^0 = [x_1, x_2, x_3]^T \). To obtain \( m(\alpha^0 \times h) \), compute first \( (\alpha^0 \times u) \oplus f(x, u, w) = [x_1 x_2 + u + x_1, \text{sign}(x_1 + 1) + x_4]^T \). Then, \( m([x_1 x_2]) = [x_1, x_2]^T \) and \( a^1 = [x_1, x_2]^T \). It can be shown in the similar manner that \( \alpha^2 = \alpha^1 = [x_1, x_2]^T \). So, \( \alpha = [x_1, x_2]^T = [z_1, z_2]^T \) and \( z^+_1 = y_1 z_2 + u + y_1, \quad z^+_2 = \text{sign}(y_1 + 1) + y_2, \quad y_s = z_1 = 1 \).

According to Algorithm 2 one has \( Z_2 = \{z_1\}, \quad y_0 = \{y_2\} \). Since \( F_2(z, y, u) \) depends on \( y_2 \) but not on \( u \), \( Z_0 = \emptyset \). The function \( F_2(z, y, u) \) depending on \( z_2 \) is \( F_2(z, y, u) \) and it depends on \( u \) too, therefore insert \( z_2 \) into \( y_0 \), since \( Z_0 = \emptyset \), go to Step 4. On Step 4 one gets \( u = z_2 y_1 + v \) and on Step 5 \( \xi = z_1 \). Because \( \xi \) is obviously \( \xi \)-invariant and \( \alpha^0 \leq \alpha \leq h_0 \), the DDDPM is solved. Because \( u \) depends on \( z_2 \) (but not on \( z_1 \) and \( z_2 = x_2 \) is \( (h, f) \)-invariant, one may reduce, according to Remark 1, the dimension of the dynamics of the compensator. The dynamic compensator \( z^+_2 = \text{sign}(y_1 + 1) + y_2, \quad z_2(0) = x_2(0), \quad u = -z_2 y_1 + v \) solves the problem, since \( x^+_1 = x_1 + v \).

### 6 Conclusions

The necessary and sufficient solvability condition of the DDDPM has been given for discrete-time nonlinear control systems, not necessarily described in terms of smooth functions. Moreover, the algorithm is provided that computes the feedback.

**Acknowledgements**

The work of A. Kaldmae and Ü. Kotta was supported by the EU through European Regional Development Fund, the target funding project SF0140018s08 of Estonian Ministry of Education and Research. Additionally, A. Kaldmae was supported by the ESF grant N8787. The work of A. Shumsky and A. Zhirabok was supported by the Far Eastern Federal University.

### References

1. Institute of cybernetics at tallinn university of technology: The nonlinear control website [http://webmathematica.cc.ioc.ee/webmathematica/NNControl/funcalg].