Output Feedback Disturbance Decoupling in Discrete-Time Nonlinear Systems

Ülle Kotta ∗ Alexey Ye. Shumsky ∗∗ Alexey N. Zhirabok ∗∗∗

Abstract: The paper studies the unmeasurable disturbance decoupling problem via the dynamic output feedback for discrete-time nonlinear control systems. To address the problem the novel algebraic approach, called the algebra of functions, is applied. The advantage of the latter over the differential geometric methods is that the system description may depend on non-differentiable functions.

Keywords: Disturbance rejection, dynamic output feedback, nonlinear control systems, discrete-time systems, algebraic approaches

1. INTRODUCTION

The disturbance decoupling problem (DDP) for discrete-time nonlinear control system by state feedback has been addressed in many papers: Aranda-Bricaire and Kotta [2004, 2001], Fliege and Nijmeijer [1994], Grizzle [1985], Kotta and Nijmeijer [1991], Monaco and Normand-Cyrot [1984]. Most results extend the known results for continuous-time systems (see for example, Nijmeijer and van der Schaft [1990], Conte et al. [2007], Isidori [1995]) and all these papers deal with the systems described by the difference equations defined in terms of sufficiently smooth functions. However, there are no papers that address this problem using the output feedback except the paper by Shumsky and Zhirabok [2010]. The results of this paper provide only a partial solution to the output dynamic disturbance decoupling problem (DDDPO); namely a certain vector function of outputs will be made independent of disturbances and this vector function may have less components than the output vector itself. The goal of this paper is to extend the above results. In particular, we aim to obtain a complete algorithmic solution of the problem and prove a few propositions related to the disturbance decoupling problem solution.

Note that the algorithm we present was partly sketched in the discussion paper by Kotta and Mullari [2010]. However, the goal of the sketch was not to provide a complete algorithm with all necessary details for computation the dynamic output feedback but rather to single out the main steps of the solution of the partial DDDPO in the paper by Shumsky and Zhirabok [2010]. The focus of the latter paper was to suggest a unified approach applicable in cases of discrete- and continuous-time as well as in discrete-event systems. We repeat the more detailed version of the algorithm in this paper by several reasons. First, the proofs of Propositions 15-20 rely on this algorithm. Secondly, the algorithm in Kotta and Mullari [2010] only yields a dynamic feedback solution (when it exists) that follows from the equation \( f_0(x, y, u^c) = u^* \) at Step 6. The algorithm in this paper may also result in a static feedback solution (when it exists), that is when the last equation in Algorithm 3 below has a form \( v_i = \gamma_i(y, u) \). Finally, note that our algorithm is more general than the earlier algorithm: we may obtain a solution even when the procedure in Shumsky and Zhirabok [2010] fails. Moreover, our algorithm may yield a feedback with a lower dimension.

Like in Shumsky and Zhirabok [2010], to address the problem, the novel algebraic approach, called the algebra of functions, is applied, see Zhirabok and Shumsky [2008]. The main idea for developing the algebra of functions traces back to the book by Hartmanis and Stearns [1966], which introduced the algebra of partitions for finite automata defined via the transition tables or graphs. In the algebra of functions the partitions were replaced by functions generating them and the analogous operations and operators for functions were introduced. The advantage of the suggested approach over the conventional differential geometric methods is that the system description may depend on non-differentiable functions. The disadvantage is that the computations with the key elements of the approach are not (yet) well-formalized.

As for the continuous-time nonlinear systems there exist also only a few papers addressing the problem, see Pothin

1 Alternatively from equation \( v_i = \gamma_i(\tilde{z}, y, u) \) at Step 5 of Algorithm 3 in this paper.
et al. [2002], Isidori et al. [1981], Xia and Moog [1999], Andiarti and Moog [1996]. The paper by Pothin et al. [2002] studies the problem using a static measurement feedback, and in Isidori et al. [1981] the feedback considered is restricted to the so-called pure dynamic measurement feedback, whereas the other two papers focus on the dynamic measurement feedback. Finally, let us mention that the results have been obtained also for continuous-time nonlinear descriptor systems in Shumsky [2009].

2. PROBLEM STATEMENT

Consider a discrete-time nonlinear control system

\[ x(k + 1) = f(x(k), u(k), w(k)) \]

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 disturbance and \( y \in Y \subseteq \mathbb{R} \) is the output. The disturbance decoupling problem under a dynamic output feedback (DDDO) can be stated as follows: find a regular dynamic output feedback of the form

\[ \eta(k + 1) = g(\eta(k), y(k), v(k)) \]
\[ u(k) = \psi(\eta(k), y(k), v(k)) \]

with \( v \in V \subseteq \mathbb{R}^m \) such that the outputs of the closed-loop system for all \( k \geq 0 \) are independent of the disturbances \( w \). One says that the DDDPO is partially solvable if the independence of the disturbances is guaranteed only for a certain vector function of output that may be of lower dimension than the output vector itself. This vector function of output is not fixed a priori but the algorithm is searching for such a function with a maximal dimension. One says that the disturbance decoupling problem is solvable via static output feedback if \( u(k) = \psi(y(k), v(k)) \).

3. THE ALGEBRA OF FUNCTIONS

To address the DDDPO, the mathematical technique called the algebra of functions and developed in Zhirabok and Shumsky [2008] will be used. We recall below briefly the definitions and concepts to be used in this paper, see also Shumsky [2009].

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 arbitrary set \( S \) of vector functions 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 \( \alpha, \beta \in S \), one says that \( \alpha \leq \beta \) if there exists a function \( \gamma \) such that \( \beta(s) = \gamma(\alpha(s)) \) for all \( s \in S \).

The definition means that every component of the function \( \beta \) can be expressed as a function of \( \alpha \). Clearly, for differentiable functions \( \alpha \leq \beta \) if

\[ \text{rank}[\partial \alpha/\partial s] = \text{rank} \begin{bmatrix} \partial \alpha_1/\partial s \\ \partial \alpha_2/\partial s \end{bmatrix} \]

Example 2. Let \( \alpha(s) = (s_1, s_2) \), \( \beta(s) = (s_1, s_1 s_2) \). Then \( \alpha \leq \beta \) since there exists \( \gamma(\alpha) = (\alpha_1, \alpha_1 \alpha_2) \) such that \( \beta_1 = \alpha_1, \beta_2 = \alpha_1 \alpha_2 \). The inequality \( \beta \leq \alpha \) does not hold in general, since \( \alpha_2 = \beta_2 / \beta_1 \) is not valid for \( s_1 = \beta_1(s) = 0 \), i.e. on the set of measure zero.

Definition 3. (Strict 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 \( S \) into the equivalence classes containing the equivalent functions.

Example 4. The functions \( \alpha(s) = (s_1, s_2)^T \) and \( \beta(s) = (s_1, s_1 + s_2)^T \) are strictly equivalent since \( \beta_1 = \alpha_1, \beta_2 = \alpha_1 + \alpha_2 \), and \( \alpha_2 = \beta_1, \alpha_2 = \beta_2 - \beta_1 \).

Besides the strict equivalence, we use the notion of equivalence, corresponding to the situation when one of the inequalities \( \alpha \leq \beta \) or \( \beta \leq \alpha \) may be violated on a set of measure zero.

Example 5. (Continuation of Example 2) The functions \( \alpha \) and \( \beta \) are equivalent though not strictly equivalent.

Definition 6. Given \( \alpha, \beta \in S \),

\[ \alpha \times \beta = \max(\gamma \in S | \gamma \leq \alpha, \gamma \leq \beta), \]

and

\[ \alpha \oplus \beta = \min(\gamma \in S | \alpha \leq \gamma, \beta \leq \gamma). \]

It follows from these definitions that the function \( \alpha \times \beta \) is a maximal bottom of the functions \( \alpha \) and \( \beta \) while \( \alpha \oplus \beta \) is their minimal top. In the simple cases the latter property may be used to compute \( \alpha \oplus \beta \).

The rule for operation \( \times \) is simple \( (\alpha \times \beta)(s) = [\alpha(s), \beta(s)]^T \).

Example 7. (Computation of the functions \( \alpha \times \beta \) and \( \alpha \oplus \beta \)). Let \( S = \mathbb{R}^3 \), \( \alpha(s) = [s_1 + s_2, s_3]^T \), \( \beta(s) = [s_1 s_3, s_2 s_3]^T \). Then \( (\alpha \times \beta)(s) = [s_1 + s_2, s_1 s_3, s_1 s_3]^T \), \( (\alpha \oplus \beta)(s) = [s_3(s_1 + s_2)]^T \).

Definition 8. (Binary relation \( \Delta \)) Given \( \alpha, \beta \in S_X \), there exists a function \( f_s \) such that

\[ (\alpha, \beta) \in \Delta \iff \beta(f(x, u)) = f_s(\alpha(x), u) \]

for all \( (x, u) \in X \times U \).

When \( (\alpha, \beta) \in \Delta \), it is said that \( \alpha \) and \( \beta \) form an (ordered) pair.

The example below shows that the binary relation is not symmetric.

Example 9. Let \( \alpha(x) := x_2, \beta(x) := x_1 \), and the state transition map in (1)

\[ f = \begin{bmatrix} \varphi_1(x_2, u) \\ \varphi_2(x_1, x_2, u) \end{bmatrix}. \]

Then \( \beta(f(x, u)) = \varphi_1(\alpha(x), u) \), but

\[ \alpha(f(x, u)) = \varphi_2(x_1, x_2, u) \neq f_s(\beta(x), u). \]

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

Definition 10. Operator \( m(\alpha) \) is a function in \( S_X \) that satisfies the following conditions

\[ (i) \ (\alpha, m(\alpha)) \in \Delta \]
Definition 11. Operator $M(\beta)$ is a function in $S_X$ that satisfies the following conditions:

(i) $(\alpha, \beta) \in \Delta$, then $m(\alpha) \leq \beta$. 

(ii) if $(\alpha, \beta) \in \Delta$, then $a \leq M(\beta)$. 

Computation of the operator $m$. It has proven that the function $\gamma$ exists that satisfies the condition $(\alpha \times u) \oplus f \cong \gamma(f)$; define $m(\alpha) \cong \gamma$, see Shumsky [1988].

The examples how to compute $\gamma$ may be found in Zhirabok and Shumsky [2008]. 

Computation of the operator $M$. In the special case when the composite function $\beta(f(x, u))$ can be represented in the form

$$\beta(f(x, u)) = \sum_{i=1}^{d} a_i(x)b_i(u)$$

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 $M(\beta) := a_1 \times a_2 \times \cdots \times a_d$. 

For the general case, see Zhirabok and Shumsky [2008].

4. PROBLEM SOLUTION

Find 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 $w$, i.e.

$$\frac{\partial}{\partial w} \alpha^0(f(x, u, w)) = 0.$$ 

Note that $\alpha^0(x)$ is not unique; however all the possible choices are equivalent functions. Moreover, applying the operators $m$ and $M$ to equivalent functions will yield again the equivalent functions. Therefore, the results of the Algorithms 1, 2 and 3 will be the same for different choices of $\alpha^0(x)$, up to the equivalence.

The function $\alpha \in S_X$ specifies the equivalence relation $E_\alpha$ on $X$ according to the rule:

$$\forall x, x' \in X \ (x, x') \in E_\alpha \Leftrightarrow \alpha(x) = \alpha(x').$$

Due to this, the function $\alpha$ is called invariant with respect to equivalence relation $E_\alpha$.

Definition 12. The function $\alpha$ is said to be invariant with respect to system dynamics or $f$-invariant if for arbitrary time instants $k_0$ and $k_1, k_1 > k_0, x(k_0), x'(k_0) \in X$ and every control sequence $u(k_0), u(k_0 + 1), \ldots, u(k_1)$, the following holds

$$(x(k_0), x'(k_0)) \in E_\alpha \Rightarrow (x(k_1), x'(k_1)) \in E_\alpha$$

where $x(k_1)$ and $x'(k_1)$ are the states at the time instant $k_1$ starting from $x(k_0)$ and $x'(k_0)$ at $k_0$, respectively, under the control sequence $u(k_0), u(k_0 + 1), \ldots, u(k_1)$. 

Lemma 13. (Shumsky and Zhirabok [2010]) The function $\alpha$ is $f$-invariant iff $(\alpha, \alpha) \in \Delta$.

Extending the above notion, one may introduce the following definition.

Definition 14. The function $\varphi$ is said to be $(h, f)$-invariant if $(\varphi \times h, \varphi) \in \Delta$. 

It can be shown that this definition is a generalization for the discrete-time systems of the form (1) of the well-known definition of $(h, f)$-invariant distribution and codistribution, introduced by Isidori et al. [1981].

Algorithm 1. (Computation of the minimal $(h, f)$-invariant function $\alpha$ satisfying the condition $\alpha^0 \leq \alpha$). Given $\alpha^0$, compute recursively, using the formula below

$$\alpha^{i+1} = \alpha^i \oplus m(\alpha^i \times h)$$

the sequence of functions $\alpha^i, i \geq 1$. By Theorem 1 in [Shumsky and Zhirabok, 2010], there exists a finite $j$ such that $\alpha^{j+1} \equiv \alpha^j$, denoted by $\alpha^{j+1} \equiv \alpha^j$. Define $\alpha := \alpha^j$. 

Algorithm 2. (Computation of the maximal $(h, f)$-invariant function $\beta$ satisfying the condition $\beta \leq \beta^0$).

Step 1. Set $i := 0$.

Step 2. Compute the function $\gamma^i = M(\beta^i)$.

Step 3. If the components of the vector function $\gamma^i$ can be expressed in terms of the components of the function $h \times \beta^0 \times \beta^1 \times \cdots \times \beta^i$, then go to Step 5. Otherwise, go to Step 4.

Step 4. Find the vector function $\beta^{i+1}$ with minimal numbers of components satisfying the condition $h \times \beta^0 \times \beta^1 \times \cdots \times \beta^{i+1} \leq \gamma^i$, set $i := i + 1$ and go to Step 2.

Step 5. Define $\beta := \beta^0 \times \beta^1 \times \cdots \times \beta^i$.

The Algorithm 3 below provides the partial solution of the DDDPO whenever it exists.

Algorithm 3. 

Step 1. Given $\alpha^0$, find, by Algorithm 1, the minimal $(h, f)$-invariant function $\varphi$ satisfying the condition $\alpha^0 \leq \varphi$. Since $\varphi$ is $(h, f)$-invariant, i.e. $(\varphi \times h, \varphi) \in \Delta$, then by Definition 8 there exists a function $F$ such that

$$\varphi(f(x, u, w)) = F(\varphi(x), h(x, u)).$$

Because $\alpha^0 \leq \varphi$, i.e. $\varphi = \gamma(\alpha^0)$ for some $\gamma$, then from the definition of $\alpha^0$, $\partial \varphi(f(x, u, w))/\partial w = 0$, i.e. $F$ does not depend on $w$. Define the function $z := \varphi(x) : X \rightarrow Z$, and construct the system

$$(z(k + 1) = \varphi(f(x(k), u(k), u(k))) = F(z(k), y(k), u(k))).$$

Step 2. If some of the components $F_j$ of the vector function $F$ depend only on $y_j = h_j(x)$ but not on $u$, then check whether $\varphi \oplus h \leq h_j$, i.e. whether $y_j$ depends on $u$. If the inequality holds, go Step 4 with $\tilde{z} := z, \tilde{\varphi} := \varphi$, and $F := F$. Otherwise exclude the corresponding components from $\varphi$, and denote the obtained vector function by $\varphi^0$.

Step 3. Repeat the Step 1 for $\varphi^0$. That is, find the minimal $(h, f)$-invariant function $\tilde{\varphi}$ satisfying the condition $\varphi^0 \leq \tilde{\varphi}$. Define the function $\tilde{z} := \tilde{\varphi}(x) : X \rightarrow \tilde{Z}$ and construct the system $\tilde{z}(k + 1) = \tilde{\varphi}(f(x(k) + 1)) = \tilde{F}(\tilde{z}(k), y(k), u(k))$. If $\tilde{\varphi} \oplus h = \text{const}$, then stop (the DDDPO is not solvable). Otherwise, go to Step 4.

Step 4. Find in the function $\tilde{F}$ all terms in the form $\gamma_i(z, y, u), i = 1, \ldots , r$, which do not contain the variable
\[ y_j = h_j(x) \] such that \( \tilde{\phi} \oplus h \leq h_j; \) \( r \) is the number of such terms. Note that not necessarily (though it may happen for some \( i \)) that \( \gamma_i = \tilde{F}_i. \)

**Step 5.** Define

\[ v_i = \gamma_i(z, y, u), \quad i = 1, \ldots, r. \]  \( (4) \)

**Step 6.** To check solvability of the system of nonlinear equations \( (4) \), assume that \( \text{rank}(\partial \gamma / \partial u) = q \) for all \( z \in \mathbb{Z}, \ y \in Y, \ u \in U \) except perhaps on a set of measure zero, where \( \gamma = [\gamma_1, \ldots, \gamma_r]^T, \) and the function \( \gamma \) contains \( m' \), components of the vector \( u \) as its arguments, \( m' \leq m. \) By analogy with Shumsky and Zhirabok [2010], consider separately three cases.

(i) \( m' = r = q; \) in this case the system of equations \( (4) \) is solvable for some \( m' \) components of the control vector \( u \) (without loss of generality suppose that they are \( u_1, \ldots, u_{m'}): \)

\[ u_j = \alpha_j(z, y, v), \quad j = 1, \ldots, m'. \]  \( (5) \)

Set \( u_j = v_j := \alpha_j(z, y, v), \ j = m' + 1, \ldots, m. \)

(ii) \( m' > r = q; \) in this case the function \( \gamma \) contains \( m' - r \) redundant components of the vector \( u. \) Without loss of generality assume that these components are the last \( m' - r \) ones, i.e. \( u_{r+1}, \ldots, u_{m'}. \) Using additional equations for these components

\[ u_i(k) = v_i(k), \quad i = r + 1, \ldots, m, \]  \( (6) \)

one can solve the system of equations \( (4) \) in the form \( (5) \) for \( j = 1, \ldots, r. \)

(iii) \( m' > r > q; \) in this case find the matrix \( P \) with \( q \) rows such that

\[ \text{rank} \left( P \frac{\partial \gamma}{\partial u} \right) = q \]

for all \( z \in \mathbb{Z}, \ y \in Y, \ u \in U \) except perhaps on a set of measure zero. The matrix \( P \) collects \( q \) functionally independent components from all components of the function \( \gamma. \) The redundant components \( u_{q+1}, \ldots, u_{m'} \) are now in the function \( P_\gamma. \) Using equation \( (6) \) for \( i = q + 1, \ldots, m, \) one can solve the equation \( v = P_\gamma \) in the form \( (5) \) for \( j = 1, \ldots, q. \)

**Step 7.** Replace, according to \( (5), \) all inputs \( u_j \) in the equations \( \dot{z}^+ = \tilde{F}(z, y, u) \) with \( \alpha_j(\tilde{\phi}, y, v). \) If the equations obtained that way do not contain the variable \( y_j = h_j(x) \) such that \( \tilde{\phi} \oplus h \leq h_j, \) go to Step 8; if some components \( \tilde{F}_j \) of the vector function \( \tilde{F} \) contain such variable and depend also on the control \( u, \) then go to Step 4, otherwise to Step 2.

**Step 8.** Check, whether \( \tilde{\phi} \leq h. \) If \( \tilde{\phi} \leq h, \) then the original system is decomposed already (see Proposition 16). If \( \tilde{\phi} \leq h \) is not valid, go to Step 9.

**Step 9.** Check, whether \( \tilde{\phi} \oplus h \neq \text{const}. \) If the condition is satisfied, the DDDPO is partially solvable via the feedback of the form

\[ \dot{z}(k+1) = \tilde{F}(\bar{z}(k), y(k), u(k)), \]
\[ u(k) = \alpha(\tilde{z}(k), y(k), u(k)) \]  \( (7) \)

for the functions \( \beta(h) \) and \( H(\tilde{\phi}) \) such that

\[ \tilde{\phi} \oplus h = \beta(h) = H(\tilde{\phi}). \]

**Remark 1.** Some functions \( \alpha_i \) in \( (5) \) may not depend on the variable \( \bar{z}; \) the corresponding expressions give the static part of the output feedback. In order to minimize the dimension of the dynamic feedback \( (2) \) (or \( (7) \)), collect all the components of \( \bar{z} \) contained on the right-hand side of \( (5), \) and denote them by \( \bar{z}' = \rho(\bar{z}). \) Find by Algorithm 2 the maximal \((h, f), \) -invariant function \( \rho \) satisfying the condition \( \rho \leq \rho \) (this function always exists: in the worst case, \( \rho = \tilde{\phi}. \) Denote \( \bar{z}' = \rho(\bar{z}) \) and construct the system \( \bar{z}'(k+1) := \rho(\tilde{F}(\bar{z}(k), y(k), u(k))) := F'(\bar{z}'(k), y(k), u(k)). \) This system defines the feedback of the minimal dimension.

**Proposition 15.** The necessary condition for solvability of the DDDPO is

\[ \tilde{\phi} \leq h. \]

**Proof.** If the DDDPO is solvable for system \( (1), \) the output \( y \) of the system must be computable on the basis of the state vector \( \bar{z}. \) Then the function \( H \) has to exist such that \( y = H(\bar{z}). \) Because \( y = h(x) = H(\tilde{\phi}(x)), \) the \( \tilde{\phi} \leq h \) by the definition of the partial preorder \( \leq. \)

**Proposition 16.** If \( \tilde{\phi} \leq h, \) the original system is decoupled already.

**Proof.** Since \( \tilde{\phi} \) is \((h, f), \) -invariant, then \((\tilde{\phi} \times h, \tilde{\phi}) \in \Delta. \) The inequality \( \tilde{\phi} \leq h \) gives \( \tilde{\phi} \times h \equiv \tilde{\phi} \) and \((\tilde{\phi}, \tilde{\phi}) \in \Delta. \) This means that \( \tilde{\phi} \) is h-invariant and the system \( \dot{z}^+ = \tilde{F}(\bar{z}, y, u) \) may be rewritten without output \( y, \) i.e. \( \dot{z}^+ = \tilde{F}(\bar{z}, u). \) Therefore, the original system is decoupled already.

**Proposition 17.** The necessary condition for partial solvability of the DDDPO is

\[ \tilde{\phi} \oplus h \neq \text{const}. \]

**Proof.** According to the definition of the partial disturbance decoupling there exists a vector function \( \beta \) of output \( y, \beta(y), \) for which the DDDPO is solvable. Therefore, there exists a function \( H(\bar{z}) \) (see the proof of Proposition 15) such that \( \beta(y) = H(\bar{z}), \) or alternatively,

\[ \beta(h(x)) = H(\tilde{\phi}(x)). \]

The latter means that inequality \( \tilde{\phi} \oplus h \neq \text{const} \) holds because otherwise the functions \( \beta \) and \( H \) are just the constant functions. If \( \tilde{\phi} \oplus h \neq \text{const}, \) one can find the functions \( H \) and \( \beta \) from the algebraic equation \( \tilde{\phi} \oplus h = H(\tilde{\phi}). \)

**Proposition 18.** If \( \partial \gamma_i / \partial \bar{z} \) (see \( (4) \)) for \( i = 1, \ldots, r \) do not depend on \( y \) and \( u \) and the DDDPO is partially solvable, then it is solvable via the static output feedback.

**Proof.** Under the condition of the proposition the variable \( \bar{z} \) is additively included into the function \( \gamma_i. \) Therefore, it can be removed from \( \gamma_i \) without violation the solvability condition.

From Proposition 18 one can immediately recast the result for linear systems.

**Proposition 19.** If the DDDPO is partially solvable for linear system, then it is partially solvable via the static output feedback.

**Proposition 20.** The following statements are equivalent:
(i) the system of the form (1) can be decomposed as shown in Figure 1;
(ii) the DDDPO is solvable for system (1);
(iii) the following inequalities hold:
\[ \alpha^0 \leq \tilde{\phi}, \quad (\tilde{\phi}, \tilde{\phi}) \in \Delta, \quad \tilde{\phi} \leq h. \]

Figure 1. Decomposition of system (1) into observable/unobservable subsystems

Proof. (i) → (ii) Obvious.

(ii) → (iii) Due to Proposition 15, the inequality \( \tilde{\phi} \leq h \) holds. According to Step 1 of Algorithm 1, the function \( \tilde{\phi} \) satisfies the conditions \( \alpha^0 \leq \tilde{\phi} \) and \( (\tilde{\phi} \times h, \tilde{\phi}) \subseteq \Delta \). Since \( \tilde{\phi} \leq h \), then \( \tilde{\phi} \times h \equiv \tilde{\phi} \) and \( (\tilde{\phi} \times h, \tilde{\phi}) \) \( \Delta \) reduces to \( (\tilde{\phi}, \tilde{\phi}) \subseteq \Delta \).

(iii) → (i) Since \( \alpha^0 \leq \tilde{\phi} \) and \( \alpha^0 \) is not identity function, \( \tilde{\phi} \) is not one-to-one. By the results of Zhirabok and Shumsky [2008], if \( \tilde{\phi} \leq h \), then \( \tilde{\phi} \times h \equiv \tilde{\phi} \) and \( (\tilde{\phi} \times h, \tilde{\phi}) \) is not one-to-one, then the initial system is unobservable.\(^3\) It is also known from Zhirabok and Shumsky [2008] that unobservable system can be decomposed as shown in Figure 1. Since the function \( \tilde{\phi} \) specifies the observable subsystem, then
\[ f^1(x^1, u) := \tilde{\phi}(f(x, u, w)) \]
describes the dynamics of the observable subsystem, with \( x^1 \) being its state. Due to the inequality \( \alpha^0 \leq \tilde{\phi} \),
\[ \partial \tilde{\phi}(f(x, u, w))/\partial w = \partial f^1(x^1, u)/\partial w = 0 \]
holds, so the disturbance \( w \) does not influence the observable subsystem.

5. EXAMPLES

To simplify the exposition, we use the symbols \( x^+ \) and \( x \) to denote \( x(k + 1) \) and \( x(k) \), respectively, and use the similar notations for the other variables as well.

Example 21. Consider the control system
\[
x^+ = \begin{bmatrix} 
\theta_1 x_1 x_1 + x_1 + \theta_1 u_1 + \theta_3 u_2 + w \\
\theta_2 x_1 x_2 + x_2 + \theta_6 u_1 - \theta_6 u_2 + \theta_7 u_3 \\
\theta_4 x_1 x_2 + \theta_4 x_1 x_2 + x_3 + \theta_8 u_3 \\
\theta_9 x_3 + x_4 \\
\theta_{10} x_1 x_2 + x_5
\end{bmatrix},
\]
\[ y = \begin{bmatrix} x_1 \\
x_4 \end{bmatrix}. \tag{8} \]

The equations (8) constitutes a simplified sampled-data model of the underwater vehicle moving on a vertical plane, and developed under the assumptions of small \( x_1 \) and \( x_2 \) values, see Shumsky [2006]. Model variables have the following meaning: \( x_1 \) is the velocity, \( x_2 \) is the angle of the trajectory, \( x_4 \) and \( x_5 \) are the trim and its time derivative, respectively, and \( x_3 \) is the depth. Model coefficients \( \theta_1 : \theta_{10} \) characterize the masses, inertia and the structural features of the vehicle. The inputs \( u_1, u_2, \) and \( u_3 \) are the forces of the upper and bottom stern thrusters and the vertical bow thruster, respectively.

Find a minimal vector function \( \alpha^0(x) \), satisfying the condition (3):
\[ \alpha^0(x) = [x_2, x_3, x_4, x_5]^T. \]

Because \( \alpha^0 \times h = id_X \) and \( m(id_X) = id_X \), then \( \alpha^0 \oplus m(h \times \alpha^0) = \alpha^0 \) and \( \alpha^1 = \alpha^0 \). Therefore \( \varphi(x) = \alpha^0(x) = [x_2, x_3, x_4, x_5]^T \) and \( z = [z_1, z_2, z_3, z_4]^T = \varphi(x) \), yielding
\[ z^+ := F(z, y, u) = \begin{bmatrix} \theta_2 y_1 z_1 + \theta_6 y_1 u_1 - \theta_6 y_2 u_2 + \theta_7 y_3 u_3 \\
\theta_3 y_2 + \theta_4 y_1 z_1 + z_2 + \theta_8 u_3 \\
\theta_9 y_2 + y_1 \\
\theta_{10} y_1 z_1 + z_4 \end{bmatrix}. \tag{9} \]

Obviously, the expression \( F_4 = \theta_{10} y_1 z_1 + z_4 \) contains the variable \( y_1 \), does not contain \( u \) and the inequality \( \varphi \oplus h \leq h_1 \) does not hold. Therefore, the component \( \varphi_4 \) must be excluded from \( \varphi \). Because the variables \( z_1 \) and \( z_3 \) do not depend on \( \varphi_4 \), one can define \( \tilde{\varphi}(x) = \varphi^0(x) \) and \( \tilde{z} = [z_1, z_2, z_3]^T = \tilde{\varphi}(x) \). Because \( \varphi \oplus h = h_2 \) holds, the variable \( y_2 \) does not contribute for the output feedback.

Since \( \tilde{\varphi} \) holds \( \leq h_0 \) does not hold, then, according to Proposition 15, the DDDPO is partially solvable. However, since \( \alpha^0 \leq \tilde{\varphi} \) is not identity function, \( \tilde{\varphi} \) is not one-to-one. By the results of Zhirabok and Shumsky [2008], if \( \tilde{\varphi} \leq h_0 \), \( (\tilde{\varphi}, \tilde{\varphi}) \subseteq \Delta \) and \( \tilde{\varphi} \) is not one-to-one, then the initial system is unobservable.

Note that
\[ q = \text{rank} \begin{bmatrix} \partial \varphi / \partial u \\
\theta_5 y_1 / \theta_6 y_1 \\
\theta_5 y_1 / \theta_6 y_1 \\
\theta_5 y_1 / \theta_6 y_1 
\end{bmatrix} = \begin{bmatrix} \theta_5 y_1 / \theta_6 y_1 \\
0 \\
0 \\
0 
\end{bmatrix} \]
equals 3 for all \( y_1 \) values except \( y_1 = 0 \). Because \( m' = m = 3 \) and \( m' = r = q \), we have the case (i), where the equations above are solvable for \( u_1, u_2, \) and \( u_3 \). Define
\[ u_1 = \frac{y_1}{\theta_6} (v_1 - \theta_2 y_1 z_1 - \theta_7 y_3 u_3) \\
u_2 = \frac{\theta_6}{\theta_2 y_1} \\
u_3 = \frac{\theta_2 y_1}{\theta_8}. \tag{10} \]

Since \( \tilde{\varphi} \oplus h = x_4 \neq \text{const} \), then according to Step 8 of Algorithm 3 \( \beta(h) = h_2 \) and \( H(\tilde{z}) = z_3 \). Using Algorithm
\[ z^\dagger = z_1 + v_1 - v_2. \] (11)

The feedback (10), (11) makes the second component of the output, i.e. \( x_4 \), independent of disturbance.

**Example 22.** Consider the control system
\[
x^T = \begin{bmatrix}
-2x_4^2u_1 + \text{sign}(1 + x_4) \\
x_3 + 2x_5 + w \\
x_4 - x_2 - x_1 - w \\
-x_4 + x_2 \\
x_4 + x_1 + w
\end{bmatrix},
\]
y = \begin{bmatrix} x_1 \\ x_3 + x_5 \\ x_4 \end{bmatrix}.

Find a minimal vector function \( \alpha^0(x) \) satisfying the condition (3): \( \alpha^0 = [x_1, x_2 - x_3, x_3 + x_5, x_4]^T \). It can be shown that \( \alpha^0(x) = \alpha(x) \), then \( z = \varphi(x) = [x_1, x_2 - x_3, x_3 + x_4]^T \), yielding
\[
z^T := F(z, y, u) = F(z, u) = \begin{bmatrix}
-z_1^2u_1 + \text{sign}(1 + z_4) \\
-z_1 - z_3 \\
z_4 - z_2 \\
-z_4 + x_2
\end{bmatrix}. 
\]

Since \( \varphi(x) = [x_1, x_2 - x_3, x_3 + x_4]^T \leq h(x) = [x_1, x_3 + x_5, x_4]^T \), then the system is decoupled already.

**6. CONCLUSIONS**

The algorithm-based solution of the partial dynamic output feedback disturbance decoupling problem has been given for discrete-time nonlinear control systems, not necessarily described in terms of smooth functions. We do believe that the suggested approach is closely related to the algebraic geometric approach and at least for analytic (or smooth) systems it is possible to use the co-distributions of differential one-forms to formalize the computations. However, we will leave this topic for future studies. The problem statement may be also extended to the case when the system has two types of outputs – measurement outputs and those we want to make independent of the disturbances like in the papers by Andiarti and Moog [1996], Xia and Moog [1999].

**REFERENCES**


