rijksuniversiteit groningen
faculteit Wiskunde en Natuurwetenschappen

Minimal State and Canonical Realizations

Bacheloronderzoek Wiskunde

Januari 2012

Student: R.R.A Prins

Begeleider: Prof. Dr. A.J. van der Schaft

Abstract

We have constructed four factorizations for four canonical realizations. For constructing these realizations we developed a Maple program. It turns out that the construction of the factorizations for the observer canonical realization and the observability canonical realization are rather straightforward. For the controller canonical realization and controllability canonical realization it is more difficult to construct the factorization. Although still possible, we have to use here our knowledge about the observer canonical realization.

Contents

1 Introduction

There are several ways to represent differential equations. There are two which we will use in this article. First of all

$$R\left(\frac{d}{dt}\right)w(t) = 0, \quad w \in \mathbb{R}^q$$

where $R(\xi) = R_0 + R_1\xi^1 + \cdots + R_n\xi^n \in \mathbb{R}^{p\times q}[\xi]$, with $\mathbb{R}^{p\times q}[\xi]$ the space of $p \times q$ polynomial matrices in the indeterminate ξ.
Second we will use the state-space representation

$$\begin{aligned}\frac{d}{dt}x &= Ax + Bu\\ y &= Cx + Du\end{aligned}$$

where x takes values in $\mathbb{R}^n$ called the state or state vector, and w is split up into an input vector u with dimension $m \leq q$ and an output vector y with dimension $p = q - m$. Furthermore $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$.
By using the first representation we are able to construct different state maps. A state map is a differential operator $X\left(\frac{d}{dt}\right)$, corresponding to the polynomial matrix $X(\xi)$, which acting on w produces a state vector x which originates from the equation $x = X\left(\frac{d}{dt}\right)w$. In the paper called "State maps from integration by parts" by van der Schaft and Rapisarda [2] there is derived how to construct state maps by using integration by parts. The authors manage to construct a matrix $\tilde{\Pi}$ which contains almost all the coefficients of the differential equation. Using factorizations of this matrix we are able to construct states, as well as to construct minimal states. In the book "Linear Systems" by Kailath [1] four canonical realizations are shown. From these realizations we are able to derive states.
Are we able to find factorization of a matrix constructed by van der Schaft and Rapisarda[2] which will lead to the four canonical realizations mentioned by Kailath[1]? Moreover, which factorization of $\tilde{\Pi}$ belongs to which realization?

In this article we will try to obtain factorizations for the following differential equation with single input and single output,

$$y^{(n)} + p_{n-1}y^{(n-1)} + \cdots + p_1y^{(1)} + p_0y = q_{n-1}u^{(n-1)} + \cdots + q_1u^{(1)} + q_0u.$$

In Section 2 it will be shown how we obtain a minimal state by using partial integration. In Section 3, four factorizations of $\tilde{\Pi}$ will be constructed for four canonical realizations. These will be for a n-th order differential equation with single input and single output. In Section 4 we will summarize and discuss our conclusion. At the end of this work there are two appendices. Appendix A contains a worked out proof for the integration by parts we use in Section 2. In Appendix B you will find a Maple program for constructing the four canonical realizations in Section 3.

2 State Maps

In this section we will summarize the results of the paper "State maps from integration by parts" by A.J. van der Schaft and P. Rapisarda[2]. One of the main results of this paper is obtaining a minimal state by using integration by parts. A sketch of this procedure is given below.
First of all we have a differential equation in the following form

$$R\left(\frac{d}{dt}\right)w(t) = 0, \quad w \in \mathbb{R}^q,$$

where $R(\xi) = R_0 + R_1\xi^1 + \cdots + R_n\xi^n \in \mathbb{R}^{p\times q}[\xi]$, with $\mathbb{R}^{p\times q}[\xi]$ the space of $p \times q$ polynomial matrices in the indeterminate ξ.

For example, if we have the differential equation $y^{(n)} + p_{n-1}y^{(n-1)} + \cdots + p_1y^{(1)} + p_0y = q_{n-1}u^{(n-1)} + \cdots + q_1u^{(1)} + q_0u$ then

$$R_0 = \begin{pmatrix} p_0 \\ -q_0 \end{pmatrix}^T, R_1 = \begin{pmatrix} p_1 \\ -q_1 \end{pmatrix}^T, R_2 = \begin{pmatrix} p_0 \\ -q_0 \end{pmatrix}^T, \cdots, R_{n-1} = \begin{pmatrix} p_{n-1} \\ -q_{n-1} \end{pmatrix}^T, R_n = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T$$

and $w = \begin{pmatrix} y \\ u \end{pmatrix}$, with $\xi = \frac{d}{dt}$.
For this differential equation we want to compute a state-space representation,

$$\begin{aligned} \frac{d}{dt}x &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

With $x \in \mathbb{R}^n$, and $w \in \mathbb{R}^q$ is split up into an input vector u with dimension $m \leq q$ and an output vector y with dimension $p = q - m$. Furthermore $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{l\times m}$. We call x the state of the system.
It follows that state maps for the system given above can be computed from a factorization of a two-variable polynomial matrix $\Pi(\zeta, \eta)$, which has a coefficient matrix $\tilde{\Pi}$ that consists of the coefficients of the differential equation. We are able to construct $\tilde{\Pi}$ in two ways. For the first way we start with the following:
Take any n-times differentiable functions $w : \mathbb{R} \to \mathbb{R}^q$ and $\varphi : \mathbb{R} \to \mathbb{R}^q$. For each time instant $t_1 \leq t_2$ integration by part yields

$$\int_{t_1}^{t_2} w^T(t)R^T\left(-\frac{d}{dt}\right)\varphi(t)dt = \int_{t_1}^{t_2} \varphi^T(t)R\left(\frac{d}{dt}\right)w(t)dt + B_\Pi(\varphi, w)\,|_{t_1}^{t_2},$$

where $B_\Pi(\varphi, w)(t)$ "the *remainder*" is in the following form:

$$B_\Pi(\varphi, w)(t) = \begin{pmatrix} \varphi^T(t) & \varphi^{(1)T}(t) & \cdots & \varphi^{(n-1)T}(t) \cdots \end{pmatrix} \tilde{\Pi} \begin{pmatrix} w(t) \\ w^{(1)}(t) \\ \vdots \\ w^{(n-1)}(t) \\ \vdots \end{pmatrix}$$

In this equation $\varphi^{(k)} := \frac{d^k}{dt^k}\varphi$ and $w^{(k)} := \frac{d^k}{dt^k}w$, for some constant infinite matrix $\tilde{\Pi}$.
In fact, the remainder only depends on φ, w and their time derivatives up to the order $n-1$.
A proof is shown in appendix A.
$\tilde{\Pi}$ turns out to be the following matrix:

$$\tilde{\Pi} = \begin{pmatrix} -R_1 & -R_2 & \cdots & -R_{n-1} & -R_n & \cdots \\ R_2 & R_3 & \cdots & R_n & 0 & \cdots \\ \vdots & \vdots & \iddots & \iddots & \vdots & \cdots \\ (-1)^{n-1}R_{n-1} & (-1)^{n-1}R_n & 0 & \cdots & 0 & \cdots \\ (-1)^n R_n & 0 & 0 & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Because only a finite part of $\tilde{\Pi}$ contains nonzero elements, we only take this part into account and we end up with:

$$\tilde{\Pi} = \begin{pmatrix} -R_1 & -R_2 & \cdots & -R_{n-1} & -R_n \\ R_2 & R_3 & \cdots & R_n & 0 \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ (-1)^{n-1}R_{n-1} & (-1)^{n-1}R_n & 0 & \cdots & 0 \\ (-1)^n R_n & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Now if take a look at the example in the beginning of this section $\tilde{\Pi}$ turns out to be the following matrix:

$$\tilde{\Pi} = \begin{pmatrix} -p_1 & q_1 & -p_2 & q_2 & \cdots & \cdots & -p_{n-1} & q_{n-1} & -1 & 0 \\ p_2 & -q_2 & p_3 & -q_3 & \cdots & \cdots & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \iddots & \iddots & \iddots & \iddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \iddots & \iddots & \iddots & \iddots & \vdots & \vdots & \vdots \\ (-1)^{n-1}p_{n-1} & -(-1)^{n-1}q_{n-1} & (-1)^{n-1} & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ (-1)^n & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}$$

Although van der Schaft and Rapisarda do also construct the matrix $\tilde{\Pi}$ in a more complicated manner, the method described above turns out to be sufficient within the contents of this work.
There are several ways to factorize of $\tilde{\Pi}$. In the following way:

$$\tilde{\Pi} = \tilde{Y}^T \tilde{X}$$

We know that $\tilde{\Pi}$ is the coefficient matrix of $\Pi(\zeta, \eta) = Y^T(\zeta)X(\eta)$, where

$$Y^T(\zeta) = \begin{pmatrix} I_p & I_p\zeta & I_p\zeta^2 & \cdots & I_p\zeta^{n-1} \end{pmatrix} \tilde{Y^T} \text{ and } X(\eta) = \tilde{X} \begin{pmatrix} I_q \\ I_q\eta \\ I_q\eta^2 \\ \vdots \\ I_q\eta^{n-1} \end{pmatrix}$$

Now we use Theorem 2.6 [2]

Theorem 2.1. *For any factorization* $\Pi(\zeta, \eta) = Y(\zeta)^T X(\eta)$ *the map*

$$w \mapsto x := X\left(\frac{d}{dt}\right) w$$

is a state map.

If the factorization is minimal then the state map is a minimal state map. There are several factorization with this property, but in this work we will concentrate on the construction of minimal states for the four special canonical realizations mentioned in Kailath's work [1], they are mentioned below.

1 observer canonical realization

2 observability canonical realization

3 controller canonical realization

4 controllability canonical realization

3 Canonical Realizations

The textbook "Linear Systems" by Thomas Kailath [1] mentions the following canonical realizations:

1 observer canonical realization

2 observability canonical realization

3 controller canonical realization

4 controllability canonical realization

In this section we will try to obtain the factorizations of $\tilde{\Pi}$ that correspond to these realizations. Each subsection will start with a figure of the corresponding system. All of the systems are, for reasons of exposition, based on the third order differential equation:

$$y^{(3)} + p_2 y^{(2)} + p_1 y^{(1)} + p_0 y = q_2 u^{(2)} + q_1 u^{(1)} + q_0 u$$

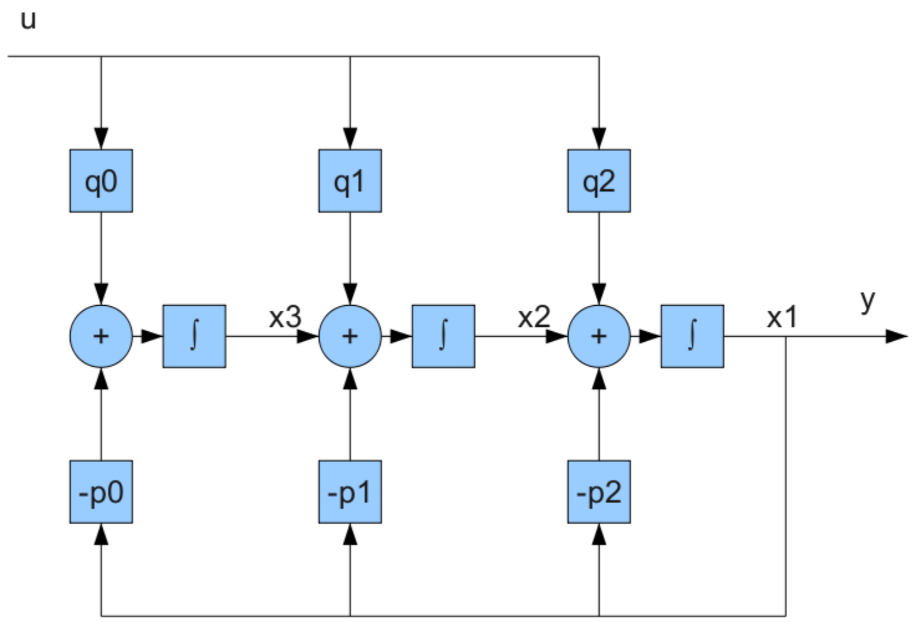
The factorizations are for all differential equations of the following form,

$$y^{(n)} + p_{n-1} y^{(n-1)} + \cdots + p_1 y^{(1)} + p_0 y = q_{n-1} u^{(n-1)} + \cdots + q_1 u^{(1)} + q_0 u$$

First we have to construct the minimal state for each of these systems. To construct the minimal states for the observer canonical form and observability canonical form we can use the figures. The controller canonical form and the controllability canonical form do however require a little more work. Thereafter we have to find for each realization the factorization of $\tilde{\Pi}$. In Subsection 1 we will start with the observer canonical realization. Then in Subsection 2 the observability canonical realization will be discussed. In Subsection 3 we have to use what we know from the observer canonical realization to compute the factorization for the controller canonical realization. In Subsection 4 we also need the observer canonical realization for the controllability canonical realization.

3.1 Observer canonical realization

Figure 1: Observer Canonical Form



In this section we are going to construct the factorization of $\tilde{\Pi}$ for the observer canonical realization. In Figure 1 the system of the observer canonical form is shown. We use this figure to calculate the minimal state. Therefore we have to start at the end of the system. If we read back we can see that $x_1 = y$. To calculate x_2 we have to differentiate x_1, add $p_2 y$ and subtract $q_2 u$. To calculate x_3 we have to differentiate x_2, add $p_1 y$ and subtract $q_1 u$.
This approach can also be used for n-th order differential equations. In that case we again have $x_1 = y$, and $x_k = \frac{d}{dt} x_{k-1} + p_{n+1-k} y - q_{n+1-k} u$ for $k = 2, \ldots, n$. This leads to the following minimal state

$$x_o = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y \\ y^{(1)} + p_{n-1} y - q_{n-1} u \\ y^{(2)} + p_{n-1} y^{(1)} - q_{n-1} u^{(1)} + p_{n-2} y - q_{n-2} \\ y^{(3)} + p_{n-1} y^{(2)} - q_{n-1} u^{(2)} + p_{n-2} y^{(1)} - q_{n-2} u^{(1)} + p_{n-3} y - q_{n-3} u \\ \vdots \\ y^{(n-1)} + p_{n-1} y^{(n-2)} - q_{n-1} u^{(n-2)} + \cdots + p_2 y^{(1)} - q_2 u^{(1)} + p_1 y - q_1 u \end{pmatrix}$$

Now we are going to construct the $n \times 2n$ matrix $\tilde{X}_o$ such that $\tilde{X}_o W = x_o$. Where the vector W consists of output function y and input function u and its derivatives up to and including the $(n-1)$-th derivative. In each row we look at how many times each of the elements of W appear in the minimal state. It turns out that,

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ p_{n-1} & -q_{n-1} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ p_{n-2} & -q_{n-2} & p_{n-1} & -q_{n-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ p_2 & -q_2 & p_3 & -q_3 & \cdots & 1 & 0 & 0 & 0 \\ p_1 & -q_1 & p_2 & -q_2 & \cdots & p_{n-1} & -q_{n-1} & 1 & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ y^{(1)} \\ u^{(1)} \\ \vdots \\ y^{(n-1)} \\ u^{(n-1)} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}.$$

In fact $\tilde{X}_o$ looks like $\tilde{\Pi}$. Especially

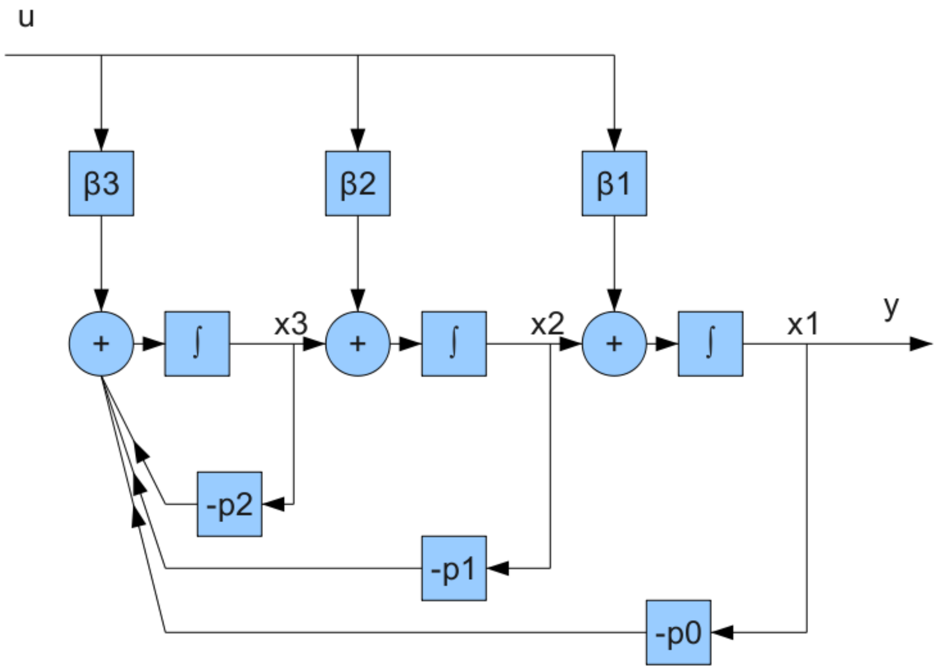
$$\tilde{X}_o = \begin{pmatrix} R_n & 0 & 0 & \cdots & 0 \\ R_{n-1} & R_n & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ R_2 & R_3 & \cdots & R_n & 0 \\ R_1 & R_2 & \cdots & R_{n-1} & R_n \end{pmatrix}.$$

The k-th row of $\tilde{X}_o$ is the same as the $(n+1-k)$-th row of $\tilde{\Pi}$. With this knowledge we are able to construct $\tilde{Y}_o^T$. Knowing that $\tilde{Y}_o^T \tilde{X} = \tilde{\Pi}$ must hold. $\tilde{Y}_o^T$ is a matrix that interchanges the rows of $\tilde{X}$ and multiplies them with -1 if necessary.
So,

$$\tilde{Y}_o^T = \begin{pmatrix} 0 & \cdots & 0 & 0 & (-1) \\ 0 & \cdots & 0 & (-1)^2 & 0 \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ 0 & (-1)^{n-1} & 0 & \cdots & 0 \\ -1^n & 0 & 0 & \cdots & 0 \end{pmatrix}$$

3.2 Observability canonical realization

Figure 2: Observability Canonical Form



In this section we are going to construct the factorization of $\tilde{\Pi}$ for the observability canonical realization. Figure 2 shows the system of the observer canonical form. In this system, the new parameters β_i are introduced. These parameters are constructed in the following way,

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ p_{n-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{n-2} & p_{n-1} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ p_2 & p_3 & \cdots & p_{n-1} & 1 & 0 \\ p_1 & p_2 & \cdots & p_{n-2} & p_{n-1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} q_{n-1} \\ q_{n-2} \\ q_{n-3} \\ \vdots \\ q_2 \\ q_1 \end{pmatrix}$$

Now we can use Figure 2 to calculate the minimal state. To do this we start at the end of the system. Reading back we see that $x_1 = y$. To calculate x_2 we differentiate x_1 and subtract $\beta_1 u$. To calculate x_3 we differentiate x_2 and subtract $\beta_2 u$.
We are also able to do this for n-th order differential equations. We again have $x_1 = y$, and $x_k = \frac{d}{dt} x_{k-1} - \beta_{k-1} u$ for $k = 2, \ldots, n$. This leads to the following minimal state:

$$x_{oa} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y \\ y^{(1)} - \beta_1 u \\ y^{(2)} - \beta_1 u^{(1)} - \beta_2 u \\ y^{(3)} - \beta_1 u^{(2)} - \beta_2 u^{(1)} - \beta_3 u \\ \vdots \\ y^{(n-1)} - \beta_1 u^{(n-2)} - \beta_2 u^{(n-3)} - \ldots - \beta_{n-2} u^{(1)} - \beta_{n-1} u \end{pmatrix}$$

We will construct the $n \times 2n$ matrix $\tilde{X}_{oa}$ such that $\tilde{X}_{oa} W = x_{oa}$. Here the vector W consists of the output y and input u and its derivatives up to and including to the $(n-1)$-th derivative. In each row we look at how many times each of the elements of W appear in the minimal state. We find the following

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\beta_1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\beta_2 & 0 & -\beta_1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & -\beta_{n-1} & 0 & -\beta_{n-2} & \cdots & 0 & -\beta_1 & 1 & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ y^{(1)} \\ u^{(1)} \\ \vdots \\ y^{(n-1)} \\ u^{(n-1)} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}.$$

So,

$$\tilde{X}_{oa} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\beta_1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\beta_2 & 0 & -\beta_1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & -\beta_{n-1} & 0 & -\beta_{n-2} & \cdots & 0 & -\beta_1 & 1 & 0 \end{pmatrix}.$$

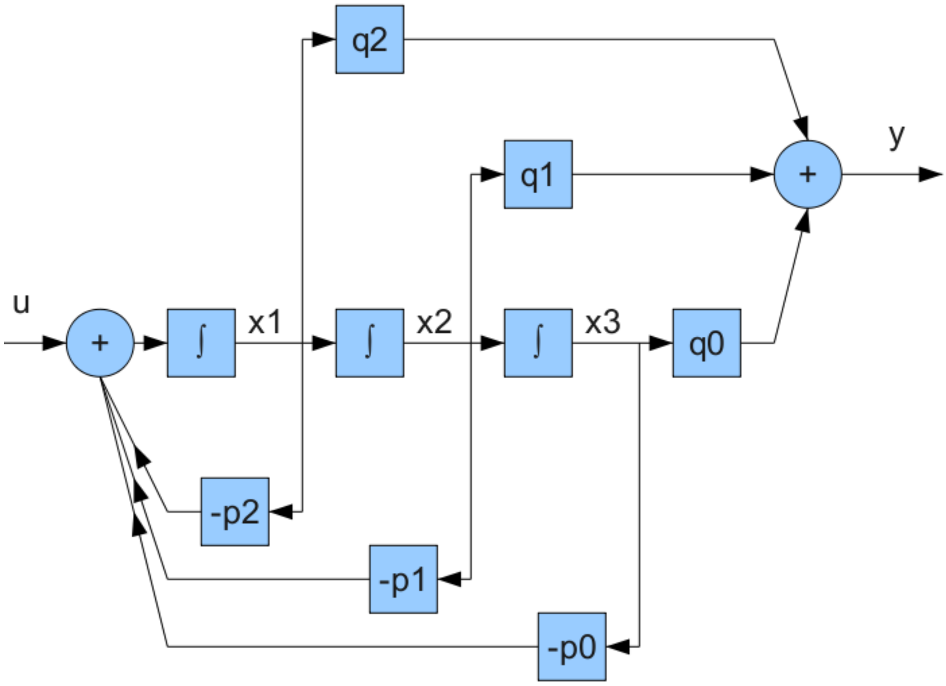
We must have that $\tilde{Y}_{oa}^T \tilde{X}_{oa} = \tilde{\Pi}$. Because of the placements of the 1's in $\tilde{X}_{oa}$ we must have that the first column of $\tilde{Y}_{oa}^T$ is the same as the first column of $\tilde{\Pi}$, the second column of $\tilde{Y}_{oa}^T$ is the same as the third column of $\tilde{\Pi}$. Especially, the k-th column of $\tilde{Y}_{oa}^T$ must be the same as the $(2k-1)$-th column of $\tilde{\Pi}$. So now $\tilde{Y}_{oa}^T$ should look as follows:

$$\tilde{Y}_{oa}^T = \begin{pmatrix} -p_1 & -p_2 & \cdots & -p_{n-2} & -p_{n-1} & 1 \\ p_2 & p_3 & \cdots & p_{n-1} & 1 & 0 \\ \vdots & \vdots & \iddots & \iddots & \iddots & \vdots \\ (-1)^{n-2} p_{n-2} & (-1)^{n-2} p_{n-1} & 1 & 0 & \cdots & 0 \\ (-1)^{n-1} p_{n-1} & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

We did not need use the β_i's in to calculate $\tilde{Y}_{oa}^T$. But if we compute $\tilde{Y}_{oa}^T \tilde{X}_{oa}$ it turns out that this is equal to $\tilde{\Pi}$.

3.3 Controller canonical realization

Figure 3: Controller Canonical Form



In this section we will construct the factorization of $\tilde{\Pi}$ for the controller canonical realization. In Figure 3 the system of the controller canonical form is shown. One can imagine how such a figure would look like for a n-th order differential equation. If we now try to construct the minimal state by reading back we get the following:

$$x_c = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} (y - q_0 x_n - q_1 x_{n-1} - q_2 x_{n-2} - \cdots - q_{n-3} x_3 - q_{n-2} x_2)/q_{n-1} \\ (y - q_0 x_n - q_1 x_{n-1} - q_2 x_{n-2} - \cdots - q_{n-3} x_3 - q_{n-1} x_1)/q_{n-2} \\ (y - q_0 x_n - q_1 x_{n-1} - q_2 x_{n-2} - \cdots - q_{n-2} x_2 - q_{n-1} x_1)/q_{n-3} \\ \vdots \\ (y - q_0 x_n - q_2 x_{n-2} - q_3 x_{n-3} - \cdots - q_{n-2} x_2 - q_{n-1} x_1)/q_1 \\ (y - q_1 x_{n-1} - q_2 x_{n-2} - q_3 x_{n-3} - \cdots - q_{n-2} x_2 - q_{n-1} x_1)/q_0 \end{pmatrix}$$

Each element of x_c depends on all the other elements of x_c. But we want to have a statement for each element in a way that it does not depend on the other elements.
The first thing I tried was to use the A_c, b_c and c_c which belong to the controller form of a system in state-space representation . Which is the following

$$\frac{d}{dt}x(t) = A_c x(t) + b_c u(t)$$
$$y(t) = c_c x(t).$$

Where,

$$A_c = \begin{pmatrix} -p_{n-1} & -p_{n-2} & -p_{n-3} & \cdots & -p_1 & -p_0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}, b_c = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

$$c_c = \begin{pmatrix} q_{n-1} & q_{n-2} & q_{n-3} & \cdots & q_1 & q_0 \end{pmatrix}.$$

We now use the fact that there exists a invertible square polynomial matrix $U(s)$ such that

$$U(s) \begin{pmatrix} sI_{n\times n} - A_c & 0_{n\times 1} & -b_c \\ -c_c & 1 & 0 \end{pmatrix} = \begin{pmatrix} I_{n\times n} & F_{n\times 1} & G_{n\times 1} \\ 0 & k & l \end{pmatrix}$$

Once we have $U(s)$, we are able to construct the minimal state in the following way:

$$x_c = Fy + Gu.$$

First we should attempt to calculate $U(s)$. $U(s)$ has the following property:

$$U(s) \begin{pmatrix} sI_{n\times n} - A_c \\ -c_c \end{pmatrix} = \begin{pmatrix} I_{n\times n} \\ 0 \end{pmatrix}$$

If we look at the special case where $y^{(2)} + p_1 y^{(1)} + p_0 y = q_1 u^{(1)} + q_0 u$ is the differential equation then we are able to construct the third row of $U(s)$. However calculating the other rows does not lead to a conclusive answer. For the next approach we use what we already know from the observer canonical realization. By using both the observer canonical realization and the controller canonical realization, we can calculate a matrix which transforms the observer canonical form into the controller canonical form. Because both the realizations are minimal, there must exist a matrix P from one realization to the other. Both systems are of the following form:

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + bu(t) \\ y(t) &= cx(t). \end{aligned}$$

Now take A_c, b_c and c_c to be the matrices for the controller system and we choose A_o, b_o and c_o for the observer system. From [3] we know P_{oc} can be calculated in the following way:

$$P_{oc} = \mathcal{C}_c {\mathcal{C}_o}^T \left(\mathcal{C}_o {\mathcal{C}_o}^T\right)^{-1}.$$

Where

$$\mathcal{C} = \begin{pmatrix} b & Ab & A^2b & A^3b & \cdots & A^{n-2}b & A^{n-1}b \end{pmatrix}.$$

P_{oc} has the following property:

$$A_c = P_{oc}A_oP_{oc}{}^{-1}, B_c = P_{oc}B_o, C_c = C_oP_{oc}{}^{-1},$$

$$x_c = P_{oc}x_o.$$

Calculating the matrix from the controller system to the observer system acquires less work so. Because of the quantities of P_{oc} we have that $P_{oc} = P_{co}{}^{-1}$. Once we have calculated P_{co} we take the inverse of this to get the matrix we need. For a third order differential equation we have the following:

$$P_{co} = \begin{pmatrix} p_1q_0 - p_0q_1 & p_2q_0 - p_0q_2 & q_0 \\ p_2q_0 - p_0q_2 & q_0 + p_2q_1 - p_1q_2 & q_1 \\ q_0 & q_1 & q_2 \end{pmatrix}.$$

We know that $P_{oc} = P_{co}{}^{-1}$ so we can construct P_{oc}. Now we are able to construct the minimal state for the controller canonical realization. We have

$$x_c = P_{oc}x_o,$$

where

$$x_o = \tilde{X}_oW.$$

So

$$x_c = P_{oc}\tilde{X}_oW.$$

Furthermore we know that

$$x_c = \tilde{X}_cW.$$

This leads to the following:

$$P_{oc}\tilde{X}_oW = \tilde{X}_cW.$$

And finally:

$$\tilde{X}_c = P_{oc}\tilde{X}_o.$$

We now have our $\tilde{X}_c$ and we need to find $\tilde{Y}_c^T$. We know that $\tilde{Y}_o^T \tilde{X}_o = \tilde{\Pi}$ so we must have the following:

$$\tilde{Y}_o^T P_{co} P_{oc} \tilde{X}_o = \tilde{Y}_o^T P_{co} \tilde{X}_c = \tilde{\Pi}$$

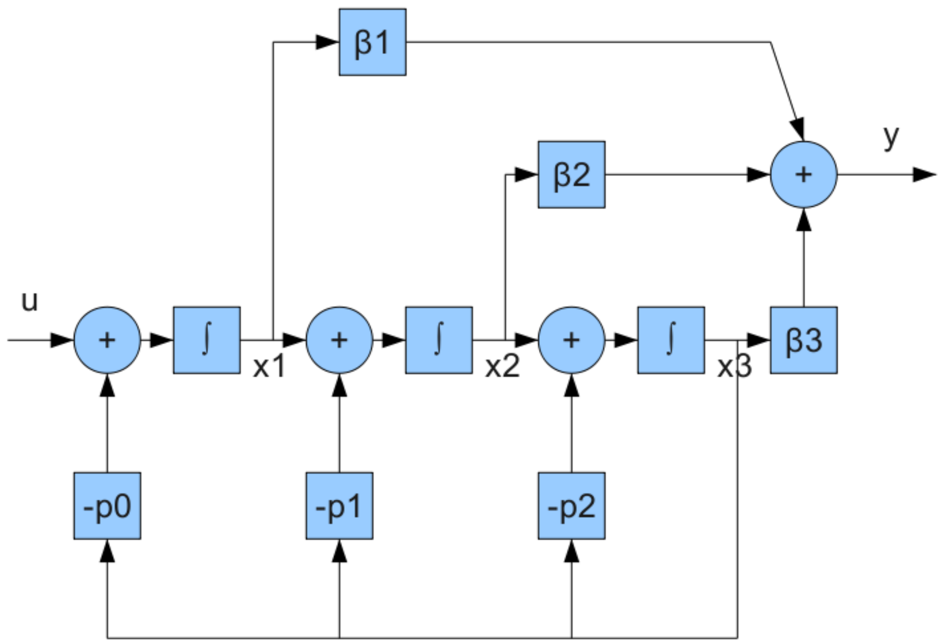
Furthermore we know that $\tilde{Y}_c^T \tilde{X}_c = \tilde{\Pi}$. Therefore:

$$\tilde{Y}_c^T = \tilde{Y}_o^T P_{co}$$

We now have the factorization of $\tilde{\Pi}$.

3.4 Controllability canonical realization

Figure 4: Controllability Canonical Form



In this section we will construct the factorization of $\tilde{\Pi}$ for the controllability canonical realization. In Figure 4 the system of the controller canonical form is shown. Again one can imagine how such a figure would look like for a n-th order differential equation. Just as for the controller canonical realization, we cannot simply read back from the figure, because then we have:

$$x_{oa} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} (y - \beta_2 x_2 - \beta_3 x_3 - \beta_4 x_4 - \cdots - \beta_{n-1} x_{n-1} - \beta_n x_n)/\beta_1 \\ (y - \beta_1 x_1 - \beta_3 x_3 - \beta_4 x_4 - \cdots - \beta_{n-1} x_{n-1} - \beta_n x_n)/\beta_1 \\ (y - \beta_1 x_1 - \beta_2 x_2 - \beta_4 x_4 - \cdots - \beta_{n-1} x_{n-1} - \beta_n x_n)/\beta_1 \\ \vdots \\ (y - \beta_1 x_1 - \beta_2 x_2 - \beta_3 x_3 - \cdots - \beta_{n-2} x_{n-2} - \beta_n x_n)/\beta_{n-2} \\ (y - \beta_1 x_1 - \beta_2 x_2 - \beta_3 x_3 - \cdots - \beta_{n-2} x_{n-2} - \beta_{n-1} x_{n-1})/\beta_1 \end{pmatrix}.$$

Again all the elements of x_{oa} depend on all the others. Therefore we have to try another way to construct the state. As we already now from the controller realization, we will try to construct a transformation matrix P which from a known system (in this case the observer canonical form) to this system (the controllability canonical form. For the controllability realization we have the following matrices

$$A_{ca} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -p_0 \\ 1 & 0 & 0 & \cdots & 0 & -p_1 \\ 0 & 1 & 0 & \cdots & 0 & -p_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & -p_{n-2} \\ 0 & 0 & \cdots & 0 & 1 & -p_{n-1} \end{pmatrix}, b_{ca} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, c_{ca} = \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_{n-1} & \beta_n \end{pmatrix}.$$

We use the controllability matrices of both systems to calculate P_{oca} in the same way as in Subsection 3.3. It turns out that it is less work to calculate P_{cao} and then take the inverse of P_{cao} to get P_{oca}. For a third order differential equation we get

$$P_{cao} = \begin{pmatrix} q_2 & -p_2q_2 + q_1 & q_2{p_2}^2 - p_1q_2 - p_2q_1 + q_0 \\ q_1 & -p_1q_2 + q_0 & q_2p_2p_1 - p_0q_2 - p_1q_1 \\ q_0 & -p_0q_2 & p_2p_0q_2 - p_0q_1 \end{pmatrix}.$$

Because we know that $P_{oca} = {P_{cao}}^{-1}$ we are able to calculate P_{oca}. Having this, we are able to calculate the minimal state for the controllability realization. We have, $x_{ca} = P_{oca}x_o$. Because we know that $x_o = \tilde{X}_oW$, so $x_c = P_{oca}\tilde{X}_oW$. Furthermore $x_{ca} = \tilde{X}_{ca}W$ so we have

$$P_{oca}\tilde{X}_oW = \tilde{X}_{ca}W.$$

From this it follows that:

$$\tilde{X}_{ca} = P_{oca}\tilde{X}_o.$$

So we have $\tilde{X}_{ca}$ and we need to find $\tilde{Y}_{ca}^T$. We know that $\tilde{Y}_o^T\tilde{X}_o = \tilde{\Pi}$ so the following must hold:

$$\tilde{Y}_o^T P_{cao}P_{oca}\tilde{X}_o = \tilde{Y}_o^T P_{cao}\tilde{X}_{ca} = \tilde{\Pi}.$$

Furthermore we know that $\tilde{Y}_{ca}^T\tilde{X}_{ca} = \tilde{\Pi}$. So:

$$\tilde{Y}_{ca}^T = \tilde{Y}_o^T P_{cao}.$$

This leads to the factorization of $\tilde{\Pi}$. We know that $\tilde{X}_o$ has full rank and P_{oca} is an invertible matrix, therefore it follows that $\tilde{X}_c$ has full rank as well. So the state map that belongs to the controllability canonical realization is minimal.

4 Conclusion

In this paper we were able to construct factorizations of the coefficient matrix for four canonical realizations for a n-th order differential equation with single input and single output. It turns out that for the observer canonical realization and the observability canonical realization these factorizations are easy to construct by first constructing the state vector. For the controller canonical realization and the controllability canonical realization we cannot use the states constructed in this way. To calculate those factorization we have to use the factorizations which were constructed for the observer. Also we managed to develop a program in Maple so that for these differential equations all the factorizations will be calculated.

Now it is of course interesting to know whether there are 'canonical' factorizations for a multiple input and multiple output system. There probably are several interesting factorizations, however these are beyond the scope of this paper and are left for future work.

References

[1] T. Kailath, "Linear Systems", Prentice Hall, Inc.,Englewood Cliffs, N.J. 07632, 1980

[2] A.J. v.d. Schaft, P. Rapisarda,"State maps from integration by parts", SIAM J. Control and Optimization, Vol. 49, No. 6, pp. 2415-2439, 2011, DOI: 10.1137/100806825

[3] P.J. Antsaklis, A.N. Michel, "A linear systems primer", Birkhaüser, Boston, 2007

A Integration

In this appendix we are going to give a proof of the following statement:

$$\int_{t_1}^{t_2} w^T(t) R^T\left(-\frac{d}{dt}\right)\varphi(t)dt = \int_{t_1}^{t_2}\varphi^T(t)R\left(\frac{d}{dt}\right)w(t)dt + B_\Pi(\varphi, w)\mid_{t_1}^{t_2}.$$

Which we used in section 2 about state maps.
First because, $R(\frac{-d}{dt}) = R_0 + R_1\frac{(-d)^1}{dt} + \cdots + R_N\frac{(-d)^n}{dt^n}$ we have that

$$\begin{aligned}
&\int_{t_1}^{t_2} w^T(t)R^T\left(-\frac{d}{dt}\right)\varphi(t)dt \\
&\qquad = \int_{t_1}^{t_2} w^T(t)R_0^T\varphi(t)dt + \int_{t_1}^{t_2} w^T(t)R_1^T\frac{-d}{dt}\varphi(t)dt \\
&\qquad + \int_{t_1}^{t_2} w^T(t)R_2^T\frac{(-d)^2}{dt^2}\varphi(t)dt + \cdots + \int_{t_1}^{t_2} w^T(t)R_n^T\frac{(-d)^n}{dt^n}\varphi(t)dt.
\end{aligned}$$

If we use for each term integral integration by parts, we get

$$\begin{aligned}
&\int_{t_1}^{t_2} w^T(t)R_0^T\varphi(t)dt \\
&\qquad = \int_{t_1}^{t_2}\left(w^T(t)R_0^T\varphi(t)\right)^T dt \\
&\qquad = \int_{t_1}^{t_2}\varphi^T(t)R_0 w(t)dt,
\end{aligned}$$

$$\begin{aligned}
&\int_{t_1}^{t_2} w^T(t)R_1^T\frac{-d}{dt}\varphi(t)dt \\
&\qquad = -\int_{t_1}^{t_2} w^T(t)R_1^T\frac{d}{dt}\varphi(t)dt \\
&\qquad = -\left(w^T(t)R_1^T\varphi(t)|_{t_1}^{t_2} - \int_{t_1}^{t_2}\frac{d}{dt}w^T(t)R_1^T\varphi(t)dt\right) \\
&\qquad = -\varphi^T(t)R_1 w(t)|_{t_1}^{t_2} + \int_{t_1}^{t_2}\varphi(t)R_1\frac{d}{dt}w(t)dt,
\end{aligned}$$

$$\begin{aligned}
&\int_{t_1}^{t_2} w^T(t)R_2^T\frac{(-d)^2}{dt^2}\varphi(t)dt \\
&\qquad = w^T(t)R_2^T\frac{d}{dt}\varphi(t)|_{t_1}^{t_2} - \int_{t_1}^{t_2}\frac{d}{dt}w^T(t)R_2^T\frac{d}{dt}\varphi(t)dt \\
&\qquad = w^T(t)R_2^T\frac{d}{dt}\varphi(t)|_{t_1}^{t_2} - \left(\frac{d}{dt}w^T(t)R_2^T\varphi(t)|_{t_1}^{t_2} - \int_{t_1}^{t_2}\frac{d^2}{dt^2}w^T(t)R_2^T\varphi(t)dt\right) \\
&\qquad = \left(\frac{d}{dt}\varphi^T(t)R_2 w(t) - \varphi^T(t)R_2\frac{d}{dt}w(t)\right)|_{t_1}^{t_2} + \int_{t_1}^{t_2}\varphi^T(t)R_2\frac{d^2}{dt^2}w(t)dt,
\end{aligned}$$

$$
\begin{aligned}
&\int_{t_1}^{t_2} w^T(t) R_3^T \frac{(-d)^3}{dt^3}\varphi(t)dt \\
&\qquad = -\left(w^T(t) R_3^T \frac{d^2}{dt^2}\varphi(t)|_{t_1}^{t_2} - \int_{t_1}^{t_2} \frac{d}{dt} w^T(t) R_3^T \frac{d^2}{dt^2}\varphi(t)dt \right) \\
&\qquad = -w^T(t) R_3^T \frac{d^2}{dt^2}\varphi(t)|_{t_1}^{t_2} + \left(\frac{d}{dt} w^T(t) R_3^T \frac{d}{dt}\varphi(t)|_{t_1}^{t_2} - \int_{t_1}^{t_2} \frac{d^2}{dt^2} w^T(t) R_3^T \frac{d}{dt}\varphi(t)dt \right) \\
&\qquad = -w^T(t) R_3^T \frac{d^2}{dt^2}\varphi(t)|_{t_1}^{t_2} + \frac{d}{dt} w^T(t) R_3^T \frac{d}{dt}\varphi(t)|_{t_1}^{t_2} - \frac{d^2}{dt^2} w^T(t) R_3^T \varphi(t)|_{t_1}^{t_2} \\
&\qquad + \int_{t_1}^{t_2} \frac{d^3}{dt^3} w^T(t) R_3^T \varphi(t)dt \\
&\qquad = \left(-\frac{d^2}{dt^2}\varphi^T(t) R_3 w(t) + \frac{d}{dt}\varphi^T(t) R_3 \frac{d}{dt} w(t) - \varphi^T(t) R_3 \frac{d^2}{dt^2} w(t) \right) |_{t_1}^{t_2} \\
&\qquad + \int_{t_1}^{t_2} \varphi^T(t) R_3 \frac{d^3}{dt^3} w(t)dt.
\end{aligned}
$$

We can see an algorithm for the integration by parts for higher derivatives,

$$
\begin{aligned}
&\int_{t_1}^{t_2} w^T(t) R_n^T \frac{(-d)^n}{dt^n}\varphi(t)dt \\
&\qquad = (-1)^n (w^T(t) R_n^T \frac{d^{n-1}}{dt^{n-1}}\varphi(t)|_{t_1}^{t_2} (-1)^{n-1} \int_{t_1}^{t_2} \frac{d}{dt} w^T(t) R_n^T \frac{d^{n-1}}{dt^{n-1}}\varphi(t)dt \\
&\qquad = (-1)^n w^T(t) R_n^T \frac{d^{n-1}}{dt^{n-1}}\varphi(t)|_{t_1}^{t_2} + (-1)^{n-1} \frac{d}{dt} w^T(t) R_n^T \frac{d^{n-2}}{dt^{n-2}}\varphi(t)|_{t_1}^{t_2} \\
&\qquad + (-1)^{n-2} \int_{t_1}^{t_2} \frac{d^2}{dt^2} w^T(t) R_n^T \frac{d^{n-2}}{dt^{n-2}}\varphi(t)dt \\
&\qquad = (-1)^n w^T(t) R_n^T \frac{d^{n-1}}{dt^{n-1}}\varphi(t)|_{t_1}^{t_2} + (-1)^{n-1} \frac{d}{dt} w^T(t) R_n^T \frac{d^{n-2}}{dt^{n-2}}\varphi(t)|_{t_1}^{t_2} \\
&\qquad + (-1)^{n-2} \frac{d^2}{dt^2} w^T(t) R_n^T \frac{d^{n-3}}{dt^{n-3}}\varphi(t)|_{t_1}^{t_2} \\
&\qquad + (-1)^{n-3} \int_{t_1}^{t_2} \frac{d^3}{dt^3} w^T(t) R_n^T \frac{d^{n-3}}{dt^{n-3}}\varphi(t)dt \\
&\qquad = (-1)^n \frac{d^{n-1}}{dt^{n-1}}\varphi^T(t) R_n w(t) + (-1)^{n-1} \frac{d^{n-2}}{dt^{n-2}}\varphi^T(t) R_n \frac{d}{dt} w(t) \\
&\qquad + \cdots - \varphi^T(t) R_n \frac{d^{n-1}}{dt^{n-1}} w(t)|_{t_1}^{t_2} \\
&\qquad + \int_{t_1}^{t_2} \varphi^T(t) R_n \frac{d^n}{dt^n} w(t)dt.
\end{aligned}
$$

Then by adding all these integrals we get:

$$\begin{aligned}
&\int_{t_1}^{t_2} w^T(t) R^T\left(-\frac{d}{dt}\right)\varphi(t)dt \\
&\qquad = \int_{t_1}^{t_2} \varphi^T(t) R_0 w(t)dt + \int_{t_1}^{t_2} \varphi^T(t) R_1 \frac{d}{dt} w(t)dt + \int_{t_1}^{t_2} \varphi^T(t) R_2 \frac{d^2}{dt^2} w(t)dt \\
&\qquad + \cdots + \int_{t_1}^{t_2} \varphi^T(t) R_{n-1} \frac{d^{n-1}}{dt^{n-1}} w(t)dt + \int_{t_1}^{t_2} \varphi^T(t) R_n \frac{d^n}{dt^n} w(t)dt + B_\Pi(\varphi, w)\mid_{t_1}^{t_2} \\
&\qquad = \int_{t_1}^{t_2} \varphi^T(t) R\left(\frac{d}{dt}\right) w(t)dt + B_\Pi(\varphi, w)\mid_{t_1}^{t_2}.
\end{aligned}$$

If we look closely at all remainder terms, we can derive the following:

$$\begin{aligned}
B_\Pi(\varphi, w) &= -\varphi^T(t)\left(R_1 w(t) + R_2 w^{(1)}(t) + \cdots + R_{n-1} w^{(n-2)}(t) + R_n w^{(n-1)}(t)\right) \\
&+ \varphi^{(1)T}(t)\left(R_2 w(t) + R_3 w^{(1)}(t) + \cdots + R_{n-1} w^{(n-3)}(t) + R_n w^{(n-2)}(t)\right) \\
&+ -\varphi^{(2)T}(t)\left(R_3 w(t) + R_4 w^{(1)}(t) + \cdots + R_{n-1} w^{(n-4)}(t) + R_n w^{(n-3)}(t)\right) \\
&+ \cdots + \\
&+ (-1)^{n-1}\varphi^{(n-2)T}(t)\left(R_{n-1} w(t) + R_n w^{(1)}(t)\right) \\
&+ (-1)^n \varphi^{(n-1)T}(t) R_n w(t).
\end{aligned}$$

Furthermore we can see that

$$B_\Pi(\varphi, w)(t) = \begin{pmatrix} \varphi^T(t) & \varphi^{(1)T}(t) & \cdots & \varphi^{(n-1)T}(t)\cdots \end{pmatrix} \tilde{\Pi} \begin{pmatrix} w(t) \\ w^{(1)}(t) \\ \vdots \\ w^{(n-1)}(t) \\ \vdots \end{pmatrix}.$$

Where:

$$\tilde{\Pi} = \begin{pmatrix} -R_1 & -R_2 & \cdots & -R_{n-1} & -R_n & \cdots \\ R_2 & R_3 & \cdots & R_n & 0 & \cdots \\ \vdots & \vdots & ⋰ & ⋰ & \vdots & \cdots \\ (-1)^{n-1}R_{n-1} & (-1)^{n-1}R_n & 0 & \cdots & 0 & \cdots \\ (-1)^n R_n & 0 & 0 & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

B Maple Program

```

#In this program, four canonical factorizations of PI-tilde will be calculated, PI-tilde is the coefficient matrix of the following N-th order differential equation,
#p0(y)+p1(dy/dt)+p2(dy/dt)^2+...+pN-1(dy/dt)^(N-1)+1(dy/dt)^(N)=q0(u)+q1(du/dt)+q2(du/dt)^2+...+qN-1(du/dt)^(N-1)+0(dy/dt)^(N). Under the heading input one should give the coefficients of the differential equation. Under the heading calculations, the calculations are done. Under the heading output one can choose what output you want to see.

#Restart & Packages:
restart;
with(LinearAlgebra):
with(linalg):
with(VectorCalculus):

#Input:
#Here the differential equation has to be submitted

#P is a row-verctor with the coefficients of the output of the differential equation,
#furthermore: p0(y)+p1(dy/dt)+p2(dy/dt)^2+...+pN-1(dy/dt)^(N-1)+1(dy/dt)^(N)

#Q is a row-vector with the coeeficients of the input of the differential equation,
#furthermore: q0(u)+q1(du/dt)+q2(du/dt)^2+...+qN-1(du/dt)^(N-1)+0(dy/dt)^(N)
P:=Matrix([p0, p1, p2, p3, 1]):
Q:=Matrix([q0, q1, q2, q3, 0]):



#Calculations:
#In here the you calculations are done,
#some of the calculation will take time and you might not need the outcome.
#Those calculation belong in the heading: permutation matrices.
#PI-Tilde
#N is the order of the differential equation,
#Pi-Tilde is the coefficient matrics:
N:=Dimension(P)[2]-1:
PIT:=Matrix(N,2*N, fill=0):
  for w from 1 by 1 to N do
      for k from w by 1 to N do
          PIT[w,2*(k-w)+1]:=subs(PIT[w,2*k-1]=((-1)^w)*P[1,k+1],PIT[w,2*k-1])
      od:
  od:
  for v from 1 by 1 to N do
      for j from v by 1 to N do
          PIT[v,2*(j-v)+2]:=subs(PIT[v,2*j]=((-1)^(v-1))*Q[1,j+1],PIT[v,2*j])
      od:
  od:
Pi_tilde:=PIT:
w:='w': k:='k': v:='v': j:='j':


#Canonical Factorization (Observer):
```

```
#The factorization of the coefficient matrics for the obeserver realization:
XTO:=Matrix(N,2*N, fill=0):
  for w from 1 by 1 to N do
      for k from 1 by 1 to 2*N do
          XTO[w,k]:=subs(XTO[w,k]=((-1)^(N+1-w))*PIT[N+1-w,k],XTO[w,k])
      od:
  od:
 X_tilde_observer:=XTO:
YTO:=Matrix(N,N, fill=0):
  for v from 1 by 1 to N do
      YTO[v,N+1-v]:=subs(YTO[v,N+1-v]=(-1)^(N-v),YTO[v,N+1-v])
  od:
 Y_tilde_observer:=YTO:
w:='w': k:='k': v:='v':
BETA:
#Beta, which we use for the obeservability and controllability realization:
PM:=Matrix(N,N, fill=0):
  for w from 1 by 1 to N do
      PM[w,w]:=subs(PM[w,w]=1,PM[w,w])
  od:
  for v from 1 by 1 to N-1 do
      for j from 1 by 1 to v do
          PM[v+1,j]:=subs(PM[v+1,j]=P[1,N-v+j],PM[v+1,j])
      od:
  od:
w:='w': v:='v': j:='j':
INVPM:=inverse(PM):
QM:=Matrix(N,1, fill=0):
  for w from 1 by 1 to N do
      QM[w,1]:=subs(QM[w,1]=Q[1,N+1-w],QM[w,1])
  od:
w:='w':
BETA:=INVPM.QM:
#Canonical Factorization (Observability):
#The factorization for the observabilty realizaton:
XTOB:=Matrix(N,2*N, fill=0):
  for w from 1 by 1 to N do
      XTOB[w,2*w-1]:=subs(XTOB[w,2*w-1]=1,XTOB[w,2*w-1])
  od:
  for v from 2 by 1 to N do
      for j from 1 by 1 to v-1 do
          XTOB[v,2*j]:=subs(XTOB[v,2*j]=(-1)*BETA[v-j,1],XTOB[v,2*j])
      od:
  od:
 X_tilde_observability:=XTOB:
w:='w': j:='j': v:='v':
YTOB:=Matrix(N,N, fill=0):
  for w from 1 by 1 to N do
      YTOB[w,N+1-w]:=subs(YTOB[w,N+1-w]=(-1)^(N-w),YTOB[w,N+1-w])
  od:
  for v from 1 by 1 to N do
      for j from 1 by 1 to N-v do
          YTOB[v,j]:=subs(YTOB[v,j]=((-1)^v)*P[1,v+j],YTOB[v,j])
      od:
  od:
w:='w': v:='v': j:='j':
 Y_tilde_observability:=YTOB:
```

```
#Canonical Realizations
#The matrices for the first oderer system of the four canonical realizations:
AO:=Matrix(N,N, fill=0):
  for w from 1 by 1 to N-1 do
      AO[w,w+1]:=subs(AO[w,w+1]=1,AO[w,w+1])
  od:
  for v from 1 by 1 to N do
      AO[v,1]:=subs(AO[v,1]=-P[1,N+1-v],AO[v,1])
  od:
 A_observer:=AO:
BO:=Matrix(N,1, fill=0):
  for j from 1 by 1 to N do
      BO[j,1]:=Q[1,N+1-j]
  od:
 B_observer:=BO:
CO:=Matrix(1,N, fill=0):
CO[1,1]:=subs(CO[1,1]=1,CO[1,1]):
 C_observer:=CO:
 A_controller:=transpose(A_observer):
 B_controller:=transpose(C_observer):
 C_controller:=transpose(B_observer):
w:='w': v:='v': j:='j':
AOB:=Matrix(N,N, fill=0):
  for w from 1 by 1 to N-1 do
      AOB[w,w+1]:=subs(AOB[w,w+1]=1,AOB[w,w+1])
  od:
  for v from 1 by 1 to N do
      AOB[N,v]:=subs(AOB[N,v]=-P[1,v],AOB[N,v])
  od:
 A_observability:=AOB:
BOB:=Matrix(N,1, fill=0):
  for j from 1 by 1 to N do
      BOB[j,1]:=BETA[j,1]
  od:
 B_observability:=BOB:
COB:=Matrix(1,N, fill=0):
COB[1,1]:=subs(COB[1,1]=1,COB[1,1]):
 C_observability:=COB:
 A_controllability:=transpose(A_observability):
 B_controllability:=transpose(C_observability):
 C_controllability:=transpose(B_observability):
w:='w': v:='v': j:='j':
#Controllability Matrices
#The controllability matrices of the four canonical realizations:
CMC:=eval(B_controller):
  for w from 1 by 1 to (N-1) do
      L:=evalm(evalm((eval(A_controller))^w).eval(B_controller));
      CMC:=simplify(Matrix([CMC,L]))
  od:
 Controller_Matrix_Controller:=CMC:
w:='w': L:='L':
CMCO:=eval(B_controllability):
  for w from 1 by 1 to (N-1) do
      L:=evalm(evalm((eval(A_controllability))^w).eval(B_controllability));
      CMCO:=simplify(Matrix([CMCO,L]))
  od:
```

```
Controller_Matrix_Controllability:=CMCO:
w:='w': L:='L':
CMO:=eval(B_observer):
  for w from 1 by 1 to (N-1) do
      L:=evalm(evalm((eval(A_observer))^w).eval(B_observer)):
      CMO:=simplify(Matrix([CMO,L]))
  od:
Controller_Matrix_Observer:=CMO:
w:='w': L:='L':
CMOB:=eval(B_observability):
  for w from 1 by 1 to (N-1) do
      L:=evalm(evalm((eval(A_observability))^w).eval(B_observability)):
      CMOB:=simplify(Matrix([CMOB,L]))
  od:
Controller_Matrix_Observability:=CMOB:
w:='w': L:='L':

#Transformation Matrices
#The transformation matrices from in the form "P_from_to"
#In here the calculations after the "#" signs might be interresting but not
    needed for further calcultions,
#If you want the calculations o be done, you should remove the "#" signs.
Pe_C_O:=simplify(CMO.(transpose(CMC)).MatrixInverse(CMC.transpose(CMC))):
Pe_O_C:=MatrixInverse(Pe_C_O):
#Pe_O_OB:=simplify(CMOB.(transpose(CMO)).MatrixInverse(CMO.transpose(CMO))):
#Pe_OB_O:=MatrixInverse(Pe_O_OB):
#Pe_CO_O:=simplify(CMO.(transpose(CMCO)).MatrixInverse(CMCO.transpose(CMCO))):
#Pe_O_CO:=MatrixInverse(Pe_CO_O):
#Pe_C_OB:=simplify(CMOB.(transpose(CMC)).MatrixInverse(CMC.transpose(CMC))):
#Pe_OB_C:=MatrixInverse(Pe_C_OB):
#Pe_CO_C:=simplify(CMC.(transpose(CMCO)).MatrixInverse(CMCO.transpose(CMCO))):
#Pe_C_CO:=MatrixInverse(Pe_CO_C):
Pe_CO_OB:=simplify(CMOB.(transpose(CMCO)).MatrixInverse(CMCO.transpose(CMCO))):
Pe_OB_CO:=MatrixInverse(Pe_CO_OB):
#Canonical Factrorization (Controller):
#The factorization of the coefficient matrics for the controller realization:
X_tilde_controller:=simplify(Pe_O_C.X_tilde_observer):
Y_tilde_controller:=simplify(Y_tilde_observer.Pe_C_O):
#Canonical Factrorization (Controllability):
#The factorization of the coefficient matrics for the controllability
    realization:
X_tilde_controllability:=simplify(Pe_OB_CO.X_tilde_observability):
Y_tilde_controllability:=simplify(Y_tilde_observability.Pe_CO_OB):
#Output:
#Here you can choose what you want to have as output, by removing the '#' in
    the row of the output you want.

#Input:
#In here The input which was given in the beginning is shown again, and also
    the order of the system.
#P;
#Order_Differential_Equation:=N;
#Coefficients_Output:=eval(P);
#Coefficients_Input:=Q;


#Matrices for first order systems:
```

```
#In here the matrices for the following system are given:
#dx/dt=A*x+B*u
#dy/dt=C*x
#A_Observer:=A_observer; B_Observer:=B_observer; C_Observer:=C_observer;
#A_Observability:=A_observability; B_Observability:=B_observability;
    C_Observability:=C_observability;
#A_Controller:=eval(A_controller); B_Controller:=eval(B_controller);
    C_Controller:=eval(C_controller);
#A_Controllability:=eval(A_controllability); B_Controllability:=eval(
    B_controllability); #C_Controllability:=eval(C_controllability);
#Coefficientmatrix and Canonical Factorizations:
#In here the coefficient matrics and its cannonical canonical factorisations
    are given:
#PI_Tilde:=Pi_tilde;
#X_Tilde_Observer:=X_tilde_observer;
#Y_Tilde_Observer:=transpose(Y_tilde_observer);
#X_Tilde_Observability:=X_tilde_observability;
#Y_Tilde_Observability:=transpose(Y_tilde_observability);
#X_Tilde_Controller:=X_tilde_controller;
#Y_Tilde_Controller:=transpose(Y_tilde_controller);
#X_Tilde_Controllability:=X_tilde_controllability;
#Y_Tilde_Controllability:=transpose(Y_tilde_controllability);
```