Table of Contents

Example 1
- Controllability
- Stabilisability
- State feedback control
- Deadbeat Control

Example 2
- Controllability
- Stabilisability
- State feedback control
- Deadbeat Control

Example 1

We will consider an example in which the data is not informative for system identification. In this example we will consider the following data and input:

```
X = [0 1 0;0 0 1];
U = [1 0];
```

We can verify that the data is not informative for system identification by using the `[bool, A, B] = isInformIdentification(X, U)`.

```
disp("Is the data informative for system identification?");
```

```
Is the data informative for system identification?
```

```
disp(isInformIdentification(X,U));
```

```
0
```

Even though the system is not uniquely identifiable, we can still infer some properties. Note that the general form for systems that can generate this data is given by:

$$\Sigma_{i/s} = \left\{ \left(\begin{pmatrix} 0 & a_1 \\ 1 & a_2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) : a_1, a_2 \in \mathbb{R} \right\}$$

```
A = [0 sym('a1');1 sym('a2')];
B = [1;0];
```

Controllability

If we construct the controllability matrix using the general form we can see that the systems are controllable.

```
disp("The rank of the reachability matrix is:")
```

```
The rank of the reachability matrix is:
```

```
disp(rank([B A*B]));
```

```
2
```

This can be verified using the `bool = isInformControllable(X)` function.

```
disp("Is the data informative for controllability?");
```

```
Is the data informative for controllability?
```

```
disp(isInformControllable(X));
```

```
1
```

Stabilisability

Since the systems are controllable, we also know that they are stabilisable. We can verify this by using the function `bool = isInformStabilisable(X)`.

```
disp("Is the data informative for stabilisation?");
```

```
Is the data informative for stabilisation?
```

```
disp(isInformStabilisable(X));
```

```
1
```

Note that just because every individual system can be stabilised, it does not imply that there exists a state feedback controller such that all systems are stable. We will seem more details about this in the next section.

State feedback control

We know that the system is controllable. Hence we might want to find a controller that stabilises all systems of the before mentioned form. To do this we will use the function `[bool, K] = StateFeedbackYalmip(X, U)`.

```
disp("Is the data informative for stabilisation by state feedback?");
```

```
Is the data informative for stabilisation by state feedback?
```

```
disp(StateFeedbackYalmip(X, U));
```

```
0
```

TODO: Why 'Succesfully solved' and no problem if conditions are not met......

As we can see there does not seem to be a controller that stabilises all systems of this form. We can verify this result by trying to find a controller that stabilises all systems of this form at the same time.

$$A + BK = \begin{pmatrix} 0 & a_1 \\ 1 & a_2 \end{pmatrix} + \begin{pmatrix} k_1 & k_2 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} k_1 & a_1 + k_2 \\ 1 & a_2 \end{pmatrix}$$

For K to be a stabilising controller, we need the eigenvalues of $A + BK$ to be inside the unit circle.

$$P_{cl}(\lambda) = \lambda^2 - (a_2 + k_1)\lambda + (a_2 - a_1 + k_2 - k_1) = 0 \Rightarrow \lambda = \frac{a_2 + k_1}{2} \pm \frac{\sqrt{(a_2 + k_1)^2 - 4 * (a_2 - a_1 + k_2 - k_1)}}{2}$$

After some calculation we can find the following condition.

$$1 > \lambda \Longleftrightarrow 1 > a_1 - k_2 + a_2 k_1$$

Lets take $a_2 \to \pm\infty$, then $k_1 \to 0$ to counteract a_2. If we also take $a_1 \to \pm\infty$ we need that $k_2 > a_1 - 1$. But since k_2 needs to be a fixed value, we can always find a bigger a_1 such that the condition does not hold. Hence there does not exist a controller that stabilises all systems described by this data at the same time.

Deadbeat Control

TODO: fix function and add to example

Example 2

We will consider an example in which the data is not informative for system identification. In this example we will consider the following data and input:

```
X = [1 1/2 -1/4;0 1 1];
U = [-1 -1];
```

We can verify that the data is not informative for system identification by using the `[bool, A, B] = isInformIdentification(X, U)`.

```
disp("Is the data informative for system identification?");
```

```
Is the data informative for system identification?
```

```
disp(isInformIdentification(X,U));
```

```
    0
```

Even though the system is not uniquely identifiable, we can still infer some properties. Note that the general form for systems that can generate this data is given by:

$$\Sigma_{i/s} = \left\{ \left(\begin{pmatrix} 1.5 + a_1 & 0.5\,a_1 \\ 1 + a_2 & 0.5 + 0.5\,a_2 \end{pmatrix}, \begin{pmatrix} 1 + a_1 \\ a_2 \end{pmatrix} \right) : a_1, a_2 \in \mathbb{R} \right\}$$

Controllability

If we construct the controllability matrix using the general form we get the following:

$$W_c = \begin{pmatrix} 1 + a_1 & (\frac{3}{2} + a_1)(1 + a_1) + \frac{a_1 a_2}{2} \\ a_2 & (1 + a_2)(1 + a_1) + \frac{a_2}{2} + \frac{a_2^2}{2} \end{pmatrix} = [B \; AB]$$

We can easily see that the matrix is not full rank for $(a_1, a_2) = (-1, 0)$. Thus since there is at least 1 system that is not controllable, the data should not be informative for controllability. This can be verified using the function `bool = isInformControllable(X)`.

```
disp("Is the data informative for controllability?");
```

```
Is the data informative for controllability?
```

```
disp(isInformControllable(X));
```

```
    0
```

Stabilisability

Since not all systems are controllable, we do not necessarily know that the data is informative for stabilisability. We can check if the data is informative for stability by using the function `bool = isInformStabilisable(X)`.

```
disp("Is the data informative for stabilisation?");
```

```
Is the data informative for stabilisation?
```

```
disp(isInformStabilisable(X));
```

```
    0
```

As we can see the data is informative for stability, hence every system of this form is stabilizable. However, this does not necessarily imply that they can be stabilised by the same controller. For this example this will be the case however, as we will see in the next section.

State feedback control

We know that the system is controllable. Hence we want to find a controller that stabilises all systems of the before mentioned form. To do this we will use the function `[bool, K] = StateFeedbackYalmip(X, U)`.

```
disp("Is the data informative for state feedback?");
```

```
Is the data informative for state feedback?
```

```
[bool, K] = StateFeedbackYalmip(X, U)
```

```
bool = logical
    1
K = 1×2
   -1.0000   -0.5000
```

As we can see there exists a K such that every system is stable. We can verify this by calculating the result:

$$A + BK = \begin{pmatrix} 1.5 + a_1 & 0.5\,a_1 \\ 1 + a_2 & 0.5 + 0.5\,a_2 \end{pmatrix} + \begin{pmatrix} 1 + a_1 \\ a_2 \end{pmatrix} \begin{pmatrix} -1 & -0.5 \end{pmatrix} = \begin{pmatrix} 0.5 & -0.5 \\ 1 & 0.5 \end{pmatrix}$$

```
disp("The eigenvalues of the closed loop system are:");
```

```
The eigenvalues of the closed loop system are:
```

```
disp(eig([0.5 -0.5;1 0.5]));
```

```
    0.5000 + 0.7071i
    0.5000 - 0.7071i
```

```
disp("The magnitude of the eigenvalues are:");
```

```
The magnitude of the eigenvalues are:
```

```
disp(abs(eig([0.5 -0.5;1 0.5])));
```

```
    0.8660
    0.8660
```

Thus we can conclude that we have found a stabilising state feedback controller for all systems of this form.

Deadbeat Control

<u>TODO: fix function and add to example</u>