A Simulative Study on Active Disturbance Rejection Control (ADRC) as a Control Tool for Practitioners

Gernot Herbst 🄪

Siemens AG, Chemnitz, Germany
gernot.herbst@siemens.com

Abstract   As an alternative to both classical PID-type and modern model-based approaches to solving control problems, active disturbance rejection control (ADRC) has gained significant traction in recent years. With its simple tuning method and robustness against process parameter variations, it puts itself forward as a valuable addition to the toolbox of control engineering practitioners. This article aims at providing a single-source introduction and reference to linear ADRC with this audience in mind. A simulative study is carried out using generic first- and second-order plants to enable a quick visual assessment of the abilities of ADRC. Finally, a modified form of the discrete-time case is introduced to speed up real-time implementations as necessary in applications with high dynamic requirements.

1   Introduction

Active disturbance rejection control (ADRC) [9, 8, 6, 7] has emerged as an alternative that combines easy applicability known from classical PID-type control methods with the power of modern model-based approaches. The foundation for ADRC is an observer who jointly treats actual disturbances and modeling uncertainties, such that only a very coarse process model is necessary in order to design a control loop, which makes ADRC an attractive choice for practitioners and promises good robustness against process variations. Present applications range from power electronics [15], motion control [14] and superconducting radio frequency cavities [16] to tension and temperature control [17].

While it can be shown (*cf*. Section 2.3) that the linear case of ADRC is equivalent to a special case of classical state space control with disturbance estimation and compensation based on the internal model principle [4], there is an important difference to model-based approaches, such as model predictive control [12] or embedded model control [2]: for the latter, an explicit model of the process to be controlled is necessary. ADRC, on the other hand, does only assume a certain canonical model regardless of the actual process dynamics and leaves all modeling errors to be handled as a disturbance. Of course this may come at the price of performance losses compared to a controller built around a precise process model or a model of the reference trajectory. Therefore, while employing the same mathematical tools, ADRC's unified view and treatment of disturbances can be seen as a certain departure from the model-based control school [13], shifting back the focus from modeling to control. This may be key to its appeal for practitioners.

The remainder of this article is organized as follows: After providing a step-by-step introduction to the linear case of ADRC in the following section, a series of simulative experiments is carried out to demonstrate the abilities of ADRC when being faced with varying process parameters or structural uncertainties and to visually provide insights into the effect of its tuning parameters. For the discrete-time case, which is introduced afterwards, an optimized formulation is presented, which enables a controller implementation with very low input-output delay.

2   Linear Active Disturbance Rejection Control

The aim of this section is to repeat and present the linear case of ADRC in a self-contained manner, following [6, 3]. While the majority of articles introduces ADRC with a second-order process, here the first-order case will be considered first and explicitly, due to its practical importance, since there are many systems—albeit technically nonlinear and higher-order—which exhibit a dominating first-order-like behavior, at least in certain operating points. The second-order case will be developed subsequently with a similar use case in mind. Additionally, it will be shown in Section 2.3 that linear ADRC can be seen as a special case from the perspective of classical state space control with disturbance compensation based on the internal model principle.

2.1 First-Order ADRC

Consider a simple first-order process, $P(s)$, with a DC gain, K, and a time constant, T:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{Ts + 1} \quad \bullet\!\!-\!\!\!-\!\!\circ \quad T \cdot \dot{y}(t) + y(t) = K \cdot u(t) \tag{1}$$

We add an input disturbance, $d(t)$, to the process, abbreviate $b = \frac{K}{T}$ and rearrange:

$$\dot{y}(t) = -\frac{1}{T} \cdot y(t) + \frac{1}{T} \cdot d(t) + \frac{K}{T} \cdot u(t) = -\frac{1}{T} \cdot y(t) + \frac{1}{T} \cdot d(t) + b \cdot u(t)$$

As our last modeling step, we substitute $b = b_0 + \Delta b$, where b_0 shall represent the known part of $b = \frac{K}{T}$ and Δb, an (unknown) modeling error, and, finally, obtain Equation (2). We will see soon that all that we need to know about our first-order process to design an ADRC is $b_0 \approx b$, *i.e.*, an approximate value of $\frac{K}{T}$. Modeling errors or varying process parameters are represented by Δb and will be handled internally.

$$\dot{y}(t) = \underbrace{\left(-\frac{1}{T} \cdot y(t) + \frac{1}{T} \cdot d(t) + \Delta b \cdot u(t) \right)}_{\text{generalized disturbance} f(t)} + b_0 \cdot u(t) = f(t) + b_0 \cdot u(t) \tag{2}$$

By combining $-\frac{1}{T} \cdot y(t)$, the disturbance $d(t)$, and the unknown part $\Delta b \cdot u(t)$ to a so-called generalized disturbance, $f(t)$, the model for our process changed from a first-order low-pass type to an integrator. The fundamental idea of ADRC is to implement an extended state observer (ESO) that provides an estimate, $\hat{f}(t)$, such that we can compensate the impact of $f(t)$ on our process (model) by means of disturbance rejection. All that remains to be handled by the actual controller will then be a process with approximately integrating behavior, which can easily be done, e.g. by means of a simple proportional controller.

In order to derive the estimator, a state space description of the disturbed process in Equation (2) is necessary:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{A} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} b_0 \\ 0 \end{pmatrix}}_{B} \cdot u(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \dot{f}(t)$$

$$y(t) = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \tag{3}$$

Since the "virtual" input, $\dot{f}(t)$, cannot be measured, a state observer for this kind of process can, of course, only be built using the input, $u(t)$, and output, $y(t)$, of the process. An estimated state, $\hat{x}_2(t)$, however, will provide an approximate value of $f(t)$, *i.e.*, $\hat{f}(t)$, if the actual generalized disturbance, $f(t)$, can be considered piecewise constant. The equations for the extended state observer (integrator process extended by a generalized disturbance) are given in Equation (4). Note that for linear ADRC, a Luenberger observer is being used, while in the original case of ADRC, a nonlinear observer was employed [7].

$$\begin{aligned} \begin{pmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} \cdot u(t) + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \cdot (y(t) - \hat{x}_1(t)) \\ &= \underbrace{\begin{pmatrix} -l_1 & 1 \\ -l_2 & 0 \end{pmatrix}}_{A - LC} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} b_0 \\ 0 \end{pmatrix}}_{B} \cdot u(t) + \underbrace{\begin{pmatrix} l_1 \\ l_2 \end{pmatrix}}_{L} \cdot y(t) \end{aligned} \tag{4}$$

One can now use the estimated variables, $\hat{x}_1(t) = \hat{y}(t)$ and $\hat{x}_2(t) = \hat{f}(t)$, to implement the disturbance rejection and the actual controller.

$$u(t) = \frac{u_0(t) - \hat{f}(t)}{b_0} \quad \text{with} \quad u_0(t) = K_{\mathrm{P}} \cdot (r(t) - \hat{y}(t)) \tag{5}$$

The according structure of the control loop is presented in Figure 1. Since K_{P} acts on $\hat{y}(t)$, rather than the actual output $y(t)$, we do have a estimation-based state feedback controller, but the resemblance to a classical proportional controller is striking to practitioners. In Equation (5), $u_0(t)$ represents the output of a linear proportional controller.
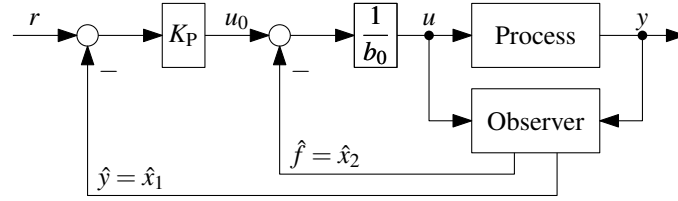
Figure 1. Control loop structure with ADRC for a first-order process.

The remainder of the control law in $u(t)$ is chosen such that the linear controller acts on a normalized integrator process if $\hat{f}(t) \approx f(t)$ holds. The effect can be seen by putting Equation (5) in Equation (2):

$$\dot{y}(t) = f(t) + b_0 \cdot \frac{u_0(t) - \hat{f}(t)}{b_0} = \left(f(t) - \hat{f}(t)\right) + u_0(t) \approx u_0(t) = K_P \cdot (r(t) - \hat{y}(t))$$

If $\hat{y}(t) \approx y(t)$ holds, we obtain a first-order closed loop behavior with a pole, $s^{CL} = -K_P$:

$$\frac{1}{K_P} \cdot \dot{y}(t) + \hat{y}(t) \approx \frac{1}{K_P} \cdot \dot{y}(t) + y(t) \approx r(t)$$

If the state estimator and disturbance rejection work properly, one has to design a proportional controller only one single time to obtain the same closed loop behavior, regardless of the parameters of the actual process. For example, one can calculate K_P from a desired first-order system with 2%-settling time:

$$K_P \approx \frac{4}{T_{\text{settle}}} \tag{6}$$

In order to work properly, observer parameters, l_1 and l_2, in Equation (4) still have to be determined. Since the observer dynamics must be fast enough, the observer poles, $s_{1/2}^{ESO}$, must be placed left of the closed loop pole, s^{CL}. A simple rule of thumb suggests for both poles:

$$s_{1/2}^{ESO} = s^{ESO} \approx (3 \ldots 10) \cdot s^{CL} \quad \text{with} \quad s^{CL} = -K_P \approx -\frac{4}{T_{\text{settle}}} \tag{7}$$

Placing all observer poles at one location is also known as "bandwidth parameterization" [6]. Since the matrix $(A - LC)$ determines the error dynamics of the observer, we can compute the necessary observer gains for the common pole location, s^{ESO}, from its characteristic polynomial:

$$\det(sI - (A - LC)) = s^2 + l_1 \cdot s + l_2 \stackrel{!}{=} \left(s - s^{ESO}\right)^2 = s^2 - 2s^{ESO} \cdot s + \left(s^{ESO}\right)^2$$

From this equation, the solutions for l_1 and l_2 can immediately be read off:

$$l_1 = -2 \cdot s^{ESO} \quad \text{and} \quad l_2 = \left(s^{ESO}\right)^2 \tag{8}$$

To summarize, in order to implement a linear ADRC for a first-order system, four steps are necessary:

1. *Modeling:* For a process with (dominating) first-order behavior, $P(s) = \dfrac{K}{Ts + 1}$, all that needs to be known is an estimate $b_0 \approx \frac{K}{T}$.

2. *Control structure:* Implement a proportional controller with disturbance rejection and an extended state observer, as given in Equations (4) and (5):

$$\begin{pmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \end{pmatrix} = \begin{pmatrix} -l_1 & 1 \\ -l_2 & 0 \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} \cdot u(t) + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \cdot y(t)$$

$$u(t) = \frac{K_P \cdot (r(t) - \hat{y}(t)) - \hat{f}(t)}{b_0} = \frac{K_P \cdot (r(t) - \hat{x}_1(t)) - \hat{x}_2(t)}{b_0}$$

3. *Closed loop dynamics:* Choose K_P, e.g. according to a desired settling time Equation (6):

$$K_P \approx \frac{4}{T_{\text{settle}}}.$$

4. *Observer dynamics:* Place the observer poles left of the closed loop pole via Equations (7) and (8):

$$l_1 = -2 \cdot s^{\text{ESO}}, \quad l_2 = \left(s^{\text{ESO}}\right)^2 \quad \text{with} \quad s^{\text{ESO}} \approx (3 \ldots 10) \cdot s^{\text{CL}} \quad \text{and} \quad s^{\text{CL}} = -K_P$$

It should be noted that the same control structure can be applied to a first-order integrating process:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K_I}{s} \quad \bullet\!\!-\!\!\circ \quad y(t) = K_I \cdot u(t)$$

With an input disturbance, $d(t)$, and a substitution, $K_I = b = b_0 + \Delta b$, with Δb representing the unknown part of K_I, we can model the process in an identical manner as Equation (2), with all differences hidden in the generalized disturbance, $f(t)$:

$$\dot{y}(t) = \underbrace{(d(t) + \Delta b \cdot u(t))}_{\text{gen. disturbance} f(t)} + b_0 \cdot u(t) = f(t) + b_0 \cdot u(t)$$

Therefore, the design of the ADRC for a first-order integrating process can follow the same four design steps given above, with the only distinction that b_0 must be set to $b_0 \approx K_I$ in step 1.

2.2 Second-Order ADRC

Following the previous section, we now consider a second-order process, $P(s)$, with a DC gain, K, damping factor, D, and a time constant, T.

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{T^2 s^2 + 2DTs + 1} \quad \bullet\!\!-\!\!\circ \quad T^2 \cdot \ddot{y}(t) + 2DT \cdot \dot{y}(t) + y(t) = K \cdot u(t) \tag{9}$$

As for the first-order case, we add an input disturbance, $d(t)$, abbreviate $b = \frac{K}{T^2}$ and split b into a known and unknown part, $b = b_0 + \Delta b$:

$$\ddot{y}(t) = \underbrace{\left(-\frac{2D}{T} \cdot \dot{y}(t) - \frac{1}{T^2} \cdot y(t) + \frac{1}{T^2} \cdot d(t) + \Delta b \cdot u(t)\right)}_{\text{generalized disturbance} f(t)} + b_0 \cdot u(t) = f(t) + b_0 \cdot u(t) \tag{10}$$

With everything else combined into the generalized disturbance, $f(t)$, all that remains of the process model is a double integrator. The state space representation of the disturbed double integrator is:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{A} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix}}_{B} \cdot u(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \dot{f}(t)$$

$$y(t) = \underbrace{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}}_{C} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} \tag{11}$$

In order to employ a control law similar to the first-order case, an extended state observer is needed to provide an estimation, $\hat{x}_1(t) = \hat{y}(t)$, $\hat{x}_2(t) = \dot{\hat{y}}(t)$ and $\hat{x}_3(t) = \hat{f}(t)$:

$$\begin{pmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \\ \dot{\hat{x}}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix} \cdot u(t) + \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \cdot (y(t) - \hat{x}_1(t))$$

$$= \underbrace{\begin{pmatrix} -l_1 & 1 & 0 \\ -l_2 & 0 & 1 \\ -l_3 & 0 & 0 \end{pmatrix}}_{A - LC} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix}}_{B} \cdot u(t) + \underbrace{\begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}}_{L} \cdot y(t) \tag{12}$$
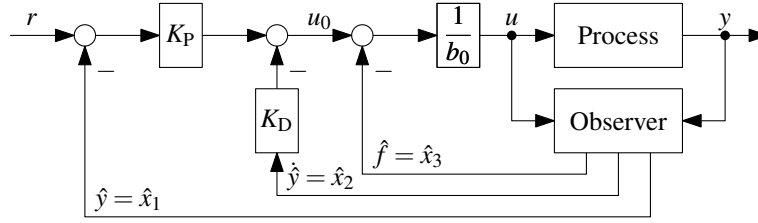
Figure 2. Control loop structure with active disturbance rejection control (ADRC) for a second-order process.

Using the estimated variables, one can implement the disturbance rejection and a linear controller for the remaining double integrator behavior, as shown in Figure 2. A modified PD controller (without the derivative part for the reference value $r(t)$) will lead to a second-order closed loop behavior with adjustable dynamics. Again, this actually is an estimation-based state feedback controller.

$$u(t) = \frac{u_0(t) - \hat{f}(t)}{b_0} \quad \text{with} \quad u_0(t) = K_P \cdot (r(t) - \hat{y}(t)) - K_D \cdot \hat{\dot{y}}(t) \tag{13}$$

Provided the estimator delivers good estimates, $\hat{x}_1(t) = \hat{y}(t) \approx y(t)$, $\hat{x}_2(t) = \hat{\dot{y}}(t) \approx \dot{y}(t)$ and $\hat{x}_3(t) = \hat{f}(t) \approx f(t)$, one obtains after inserting Equation (13) into Equation (10):

$$\ddot{y}(t) = \left(f(t) - \hat{f}(t)\right) + u_0(t) \approx u_0(t) \approx K_P \cdot (r(t) - y(t)) - K_D \cdot \dot{y}(t)$$

Under ideal conditions, this leads to:

$$\frac{1}{K_P} \cdot \ddot{y}(t) + \frac{K_D}{K_P} \cdot \dot{y}(t) + y(t) = r(t)$$

While any second-order dynamics can be set using K_P and K_D, one practical approach is to tune the closed loop to a critically damped behavior and a desired 2% settling time T_{settle}, *i.e.*, choose K_P and K_D to get a negative-real double pole, $s_{1/2}^{\text{CL}} = s^{\text{CL}}$:

$$K_P = \left(s^{\text{CL}}\right)^2 \quad \text{and} \quad K_D = -2 \cdot s^{\text{CL}} \quad \text{with} \quad s^{\text{CL}} \approx -\frac{6}{T_{\text{settle}}} \tag{14}$$

Similar to the first order case, the placement of the observer poles can follow a common rule of thumb:

$$s_{1/2/3}^{\text{ESO}} = s^{\text{ESO}} \approx (3 \dots 10) \cdot s^{\text{CL}} \quad \text{with} \quad s^{\text{CL}} \approx -\frac{6}{T_{\text{settle}}} \tag{15}$$

Once the pole locations are chosen in this manner, the observer gains are computed from the characteristic polynomial of $(A - LC)$:

$$\det\left(sI - (A - LC)\right) = s^3 + l_1 \cdot s^2 + l_2 \cdot s + l_3 \stackrel{!}{=} \left(s - s^{\text{ESO}}\right)^3 = s^3 - 3s^{\text{ESO}} \cdot s^2 + 3 \cdot \left(s^{\text{ESO}}\right)^2 \cdot s - \left(s^{\text{ESO}}\right)^3$$

The respective solutions for l_1, l_2 and l_3 are:

$$l_1 = -3 \cdot s^{\text{ESO}}, \quad l_2 = 3 \cdot \left(s^{\text{ESO}}\right)^2 \quad \text{and} \quad l_3 = -\left(s^{\text{ESO}}\right)^3 \tag{16}$$

To summarize, ADRC for a second-order system is designed and implemented as follows:

1. *Modeling:* For a process with (dominating) second-order behavior, $P(s) = \dfrac{K}{T^2 s^2 + 2DTs + 1}$, one only needs to know an approximate value $b_0 \approx \frac{K}{T^2}$.

2. *Control structure:* Implement a proportional controller with disturbance rejection and an extended state observer, as given in Equations (12) and (13):

$$\begin{pmatrix} \hat{\dot{x}}_1(t) \\ \hat{\dot{x}}_2(t) \\ \hat{\dot{x}}_3(t) \end{pmatrix} = \begin{pmatrix} -l_1 & 1 & 0 \\ -l_2 & 0 & 1 \\ -l_3 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix} \cdot u(t) + \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \cdot y(t)$$

$$u(t) = \frac{\left(K_P \cdot (r(t) - \hat{y}(t)) - K_D \cdot \hat{\dot{y}}(t)\right) - \hat{f}(t)}{b_0} = \frac{\left(K_P \cdot (r(t) - \hat{x}_1(t)) - K_D \cdot \hat{x}_2(t)\right) - \hat{x}_3(t)}{b_0}$$

3. *Closed loop dynamics:* Choose K_P and K_D, e.g. according to a desired settling time as given in Equation (14):

$$K_P = \left(s^{CL}\right)^2, \quad K_D = -2 \cdot s^{CL} \quad \text{with} \quad s^{CL} \approx -\frac{6}{T_{\text{settle}}}$$

4. *Observer dynamics:* Place the observer poles left of the closed loop poles via Equations (15) and (16):

$$l_1 = -3 \cdot s^{ESO}, \quad l_2 = 3 \cdot \left(s^{ESO}\right)^2, \quad l_3 = -\left(s^{ESO}\right)^3 \quad \text{with} \quad s^{ESO} \approx (3 \ldots 10) \cdot s^{CL}$$

2.3   Relation to Linear State Space Control with Disturbance Estimation and Compensation

Given that linear ADRC only employs tools known from classical linear state space control, how can it be compared to existing approaches? In this section, we will demonstrate that linear ADRC can be related to state space control with disturbance estimation and compensation based on the internal model principle [4]. We will start with a linear state space model of a process disturbed by $d(t)$, as follows:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t), \quad y(t) = Cx(t) \tag{17}$$

Further, we assume to possess a model for the generation of the disturbance, $d(t)$:

$$\dot{x}_d(t) = A_d x_d(t), \quad d(t) = C_d x_d(t) \tag{18}$$

Note that A_d and C_d in Equation (18) refer to modeling the disturbance generator in this section only and should not be mistaken for the discrete-time versions of A and C used in Section 4. The process model is now being extended by incorporating the internal state variables, $x_d(t)$, of the disturbance generator, resulting in an augmented process model in Equation (19), for which an observer given in Equation (20) can be set up [11]:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{x}_d(t) \end{pmatrix} = \begin{pmatrix} A & EC_d \\ 0 & A_d \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ x_d(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} \cdot u(t), \quad y(t) = \begin{pmatrix} C & 0 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ x_d(t) \end{pmatrix} \tag{19}$$

$$\begin{pmatrix} \dot{\hat{x}}(t) \\ \dot{\hat{x}}_d(t) \end{pmatrix} = \underbrace{\begin{pmatrix} A & EC_d \\ 0 & A_d \end{pmatrix}}_{\widetilde{A}} \cdot \begin{pmatrix} \hat{x}(t) \\ \hat{x}_d(t) \end{pmatrix} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\widetilde{B}} \cdot u(t) + \underbrace{\begin{pmatrix} L \\ L_d \end{pmatrix}}_{\widetilde{L}} \cdot \left(y(t) - \underbrace{\begin{pmatrix} C & 0 \end{pmatrix}}_{\widetilde{C}} \cdot \begin{pmatrix} \hat{x}(t) \\ \hat{x}_d(t) \end{pmatrix} \right) \tag{20}$$

Accordingly, the standard state space control law can now be enhanced by the estimated state variables of the disturbance generator in order to compensate or minimize the impact of the disturbance on the process, if a suitable feedback matrix, K_d, can be found:

$$u(t) = G \cdot r(t) - K \cdot \hat{x}(t) - K_d \cdot \hat{x}_d(t) \tag{21}$$

After inserting Equation (21) into Equation (17), it becomes apparent that—provided an accurate estimation, $\hat{x}_d(t)$, is available—the disturbance may be compensated to the extent that $BK_d = EC_d$ can be satisfied [11].

We will now compare the combined state and disturbance observer based on the augmented process model, as well as the control law to linear ADRC presented before. The first- and second-order case will be distinguished by (a) and (b):

- *Process model and disturbance generator:* When comparing Equations (20) to (4) and (12), respectively, one obtains the (double) integrator process with a constant disturbance model. The respective matrices E can be found using Equations (2) and (10):

  (a) $A = 0, \quad B = b_0, \quad C = 1, \quad E = \frac{1}{T}, \quad A_d = 0, \quad C_d = T = \frac{K}{b_0}$

  (b) $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ b_0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad E = \begin{pmatrix} 0 \\ \frac{1}{T^2} \end{pmatrix}, \quad A_d = 0, \quad C_d = T^2 = \frac{K}{b_0}$

- *Control law:* The comparison of Equations (5) and (21) or (13) can be made with $\hat{f} = \hat{x}_d$ being the estimated state of the disturbance generator:

(a) $\dfrac{1}{b_0} \cdot \left(K_{\mathrm{P}} \cdot (r(t) - \hat{y}(t)) - \hat{f}(t) \right) = \dfrac{K_{\mathrm{P}}}{b_0} \cdot r(t) - \dfrac{K_{\mathrm{P}}}{b_0} \cdot \hat{x}_1(t) - \dfrac{1}{b_0} \cdot \hat{f}(t)$

gives: $K_{\mathrm{d}} = \dfrac{1}{b_0}, \quad K = K_1 = \dfrac{K_{\mathrm{P}}}{b_0}, \quad G = \dfrac{K_{\mathrm{P}}}{b_0} = K_1$

(b) $\dfrac{1}{b_0} \cdot \left(K_{\mathrm{P}} \cdot (r(t) - \hat{y}(t)) - K_{\mathrm{D}} \cdot \dot{\hat{y}}(t) - \hat{f}(t) \right) = \dfrac{K_{\mathrm{P}}}{b_0} \cdot r(t) - \dfrac{K_{\mathrm{P}}}{b_0} \cdot \hat{x}_1(t) - \dfrac{K_{\mathrm{D}}}{b_0} \cdot \hat{x}_2(t) - \dfrac{1}{b_0} \cdot \hat{f}(t)$

gives: $K_{\mathrm{d}} = \dfrac{1}{b_0}, \quad K = \begin{pmatrix} K_1 & K_2 \end{pmatrix}$ with $K_1 = \dfrac{K_{\mathrm{P}}}{b_0}, K_2 = \dfrac{K_{\mathrm{D}}}{b_0}, \quad G = \dfrac{K_{\mathrm{P}}}{b_0} = K_1$

One can see that the observer and control law are equivalent in structure for both linear ADRC and a state space approach based on the internal model principle, and the model of the disturbance generator in ADRC could be made more visible by this comparison. If the standard design procedure of a state space observer and controller with disturbance compensation will lead to the same parameter values as ADRC will be verified subsequently by following the necessary steps to design K, K_{d}, G and $\widetilde{L}$ based on the same design goals used in linear ADRC before:

- *Feedback gain K*: The closed loop dynamics are determined by the eigenvalues of $(A - BK)$. For ADRC, all poles were placed on one location, s^{CL}. With this design goal, one obtains for K:

  (a) $\det(sI - (A - BK)) \overset{!}{=} s - s^{\mathrm{CL}}$ gives $K_1 = -\dfrac{s^{\mathrm{CL}}}{b_0}$

  (b) $\det(sI - (A - BK)) \overset{!}{=} \left(s - s^{\mathrm{CL}} \right)^2$ gives $K_1 = \dfrac{\left(s^{\mathrm{CL}} \right)^2}{b_0}$ and $K_2 = -\dfrac{2 s^{\mathrm{CL}}}{b_0}$

- *Gain compensation G:* In order to eliminate steady state tracking errors, G must be chosen to $G = -\left(C \cdot (A - BK)^{-1} \cdot B \right)^{-1}$, which gives $G = K_1$ for both the first- and second-order case.

- *Observer gain $\widetilde{L}$:* The dynamics of the observer for the augmented system are determined by placing the eigenvalues of $\left(\widetilde{A} - \widetilde{L}\widetilde{C} \right)$ as desired, which is the identical procedure, as in Equations (8) and (16).

- *Disturbance compensation gain K_{d}:* As mentioned above, K_{d} should be chosen to achieve $BK_{\mathrm{d}} \overset{!}{=} EC_{\mathrm{d}}$ if possible:

  (a) $BK_{\mathrm{d}} = b_0 \cdot K_{\mathrm{d}} \overset{!}{=} EC_{\mathrm{d}} = 1$ gives $K_{\mathrm{d}} = \dfrac{1}{b_0}$

  (b) $BK_{\mathrm{d}} = \begin{pmatrix} 0 \\ b_0 \cdot K_{\mathrm{d}} \end{pmatrix} \overset{!}{=} EC_{\mathrm{d}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ gives $K_{\mathrm{d}} = \dfrac{1}{b_0}$

Obviously, both designs deliver the same parameters. Based on this comparison, one may view linear ADRC and its controller design as a special case of classical state space control with an observer using a system model augmented by a certain disturbance generator model (following the internal model principle) and disturbance compensation. However, a subtle, but important, distinction has to be made: while the latter relies on a model of the plant (like all modern model-based control approaches), ADRC does always deliberately assume an integrator model for the plant and leaves all modeling errors to be handled by the disturbance estimation. Therefore, ADRC can be applied without accurately modeling the process, which presents a departure from the model-based control school [13].

3   Simulative Experiments

In the ideal case—with noise-free measurements and unlimited, ideal actuators in the control loop—ADRC would be able to suppress basically all effects of disturbances and parameter variations of the process. In practice, however, we have to live with constraints, such as limited observer dynamics or saturated controller outputs, leading to a compromise when choosing the parameters of the controller.

This section is meant to provide insights into the abilities and limitations of continuous-time ADRC in a visual manner. To that end, the controller—designed once and then left unchanged—will be confronted with a heavily varying process. The influence of the ESO pole placement (relative to the closed loop poles) will also be examined, as well as limitations of the actuator. The experiments are carried out by means of Matlab/Simulink-based simulations.

In Section 4.2, further simulations will be performed for the discrete-time implementation of ADRC, also addressing the effect of noise and sampling time.

3.1 First-Order ADRC with a First-Order Process

In a first series of experiments, a continuous-time ADRC (as introduced in Section 2.1) operating on a first-order system will be examined. The process structure and nominal parameters used to design the controller are:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{Ts + 1} \quad \text{with} \quad K = 1 \quad \text{and} \quad T = 1$$

The ADRC is designed following the four steps described in Section 2.1. For now, we assume perfect knowledge of our process and set $b_0 = \frac{K}{T} = 1$, but will leave this value unchanged for the rest of the experiments in this section. The desired closed loop settling time is chosen, $T_{\text{settle}} = 1$. To this end, the proportional gain of the linear controller is, according to Equation (6), set to $K_P = \frac{4}{T_{\text{settle}}} = 4$. Unless otherwise noted in individual experiments, the observer poles are chosen be $s_{1/2}^{\text{ESO}} = s^{\text{ESO}} = 10 \cdot s^{\text{CL}} = 10 \cdot (-K_P) = -40$. The respective values of the observer gains are obtained via Equation (8).

Throughout Section 3.1, noise-free control variables and ideal measurements will be assumed. Reaction on noisy measurements will be part of further experiments in Section 4.2.

3.1.1 Sensitivity to Process Parameter Variations

Given the explicit feature of ADRC to cope with modeling errors, our first goal will be to provide visual insights into the control loop behavior under (heavy) variations of process parameters. A series of simulations was run with fixed ADRC parameters as given above and a first-order process with varying parameters, K (DC gain) and T (time constant). In Figure 3, the closed loop step responses are displayed on the left- and the according controller outputs on the right-hand side. For both K and T, values were chosen from an interval reaching from 10% of the nominal value to 1000%, *i.e.*, a decrease and increase by factors up to ten.

The results are quite impressive. One can see that the closed loop step response remains similar or nearly identical to the desired behavior (settling time of one second) for most process parameter settings in Figure 3. Almost only for the five- and ten-fold increased time constant, larger overshoots become visible. In theory, ideal behavior (*i.e.*, almost complete ignorance of parameter variations) can be obtained by placing the observer poles far enough to the left of the closed loop poles, as will be shown in Section 3.1.2. Note that we are not constrained by actuator saturation here; this case will be examined in Section 3.1.3.

3.1.2 Effect of Observer Pole Locations

In the previous Section 3.1.1, the observer poles were placed ten times faster than the closed loop pole ($s_{1/2}^{\text{ESO}} = s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$). How does the choice of this factor influence the behavior and abilities regarding process parameter variations? We will repeat the simulations with varying time constant, T, of process both with slower and faster observers. To demonstrate a rather extreme case, as well, we will, for the faster setting, apply a factor of $s^{\text{ESO}} = 100 \cdot s^{\text{CL}}$.

The results in Figure 4 are ordered from fastest (top) to slowest observer setting (bottom). For the fastest setting, the theoretical ability to almost completely ignore any modeling error is confirmed. In order to achieve this behavior in practice, the actuator must be fast enough and must not saturate within the desired range of parameter variations. For slower observer settings, the actual closed loop dynamics increasingly differ from the desired dynamics, noticeable especially from larger overshoots for process variations with stronger low-pass character.

3.1.3 Effect of Actuator Saturation

In practice, the abilities of any controller are tied to limitations of the actuator, *i.e.*, its dynamics and the realizable range of values of the actuating variable. While the actuator dynamics can be viewed as part of the process dynamics during controller design, one has to take possible effects of actuator saturation into account.

If parameters of our example process change, the control loop behavior may be influenced or limited by actuator saturation in different ways: If the process becomes slower (*i.e.*, the time constant T increases), actuator saturation will increase the settling time. If, on the other hand, the DC gain of the process decreases, the control loop may be not be able to reach the reference value under actuator saturation.

In classical PID-type control, some sort of anti-windup strategy would be required to prevent the side-effects of actuator saturation. We will see in the experiments of this section that for ADRC, those effects can be overcome very simply by feeding the state observer with the limited actuating variable, $u_{\text{lim}}(t)$, instead of $u(t)$, either by a measured value or by an actuator model, *cf.* Figure 5.
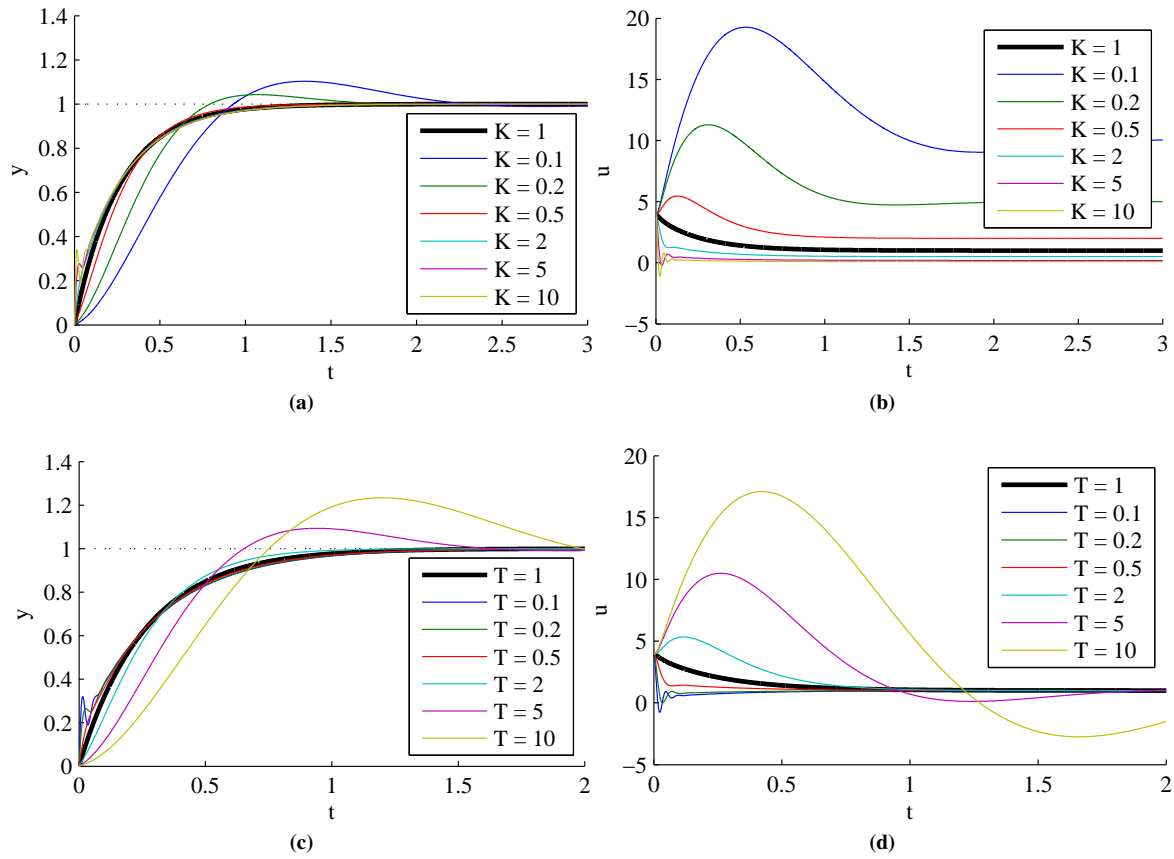
Figure 3. Experiment 3.1.1: fixed first-order ADRC controlling first-order process with varying parameters. Nominal process parameters: $K = 1$, $T = 1$. ADRC parameters: $b_0 = \frac{K}{T} = 1$, $T_{\text{settle}} = 1$, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$. (**a**) Variation of K, closed loop step response; (**b**) Controller output u for (**a**); (**c**) Variation of T, closed loop step response; (**d**) Controller output u for (**c**).

To demonstrate the behavior of ADRC under actuator saturation, the experiments of Section 3.1.1 will now be repeated under an (arbitrarily chosen) limitation of the actuating variable, $|u_{\text{lim}}| \leq 5$. Figure 6 shows the control loop behavior with varying process parameters, K and T. On the right-hand side of Figure 6, the controller outputs are shown before being fed into the actuator model, *i.e.*, before the limitation becomes effective. One can see that for reduced process gains, $K \leq 0.2$, the reference value cannot be reached anymore. From the respective controller outputs, it becomes apparent that $u(t)$ does not wind up when actuator saturation takes effect, but converges to a steady-state value.

For slower process dynamics, $T = 5$ and $T = 10$, the settling time increases considerably, yet there is almost no overshoot visible. Since the actuator is saturated, this already is the fastest possible step response. Obviously, the controller recovers very well from periods of actuator saturation. To summarize, for practical implementations of ADRC, this means that apart from the small modification shown in Figure 5, there are no further anti-windup measures necessary.

3.1.4 Effect of Dead Time

Many practical processes with dominating first-order behavior do exhibit a dead time. While there are many specialized model-based approaches to control such processes, we are interested in how ADRC will handle an unknown—albeit small—amount of dead time in the control loop.

In Figure 7, simulations were performed as in previous experiments, and a dead time, T_{dead}, was added to the process with $T_{\text{dead}} \leq 0.1$, *i.e.*, up to 10 %, compared to the time constant, T, of the process. As expected, oscillations are inevitably starting to appear with increasing T_{dead}, especially in the controller output. However, this situation can be improved if the dead time of the process is—at least approximately—known. An easy way of incorporating small dead times into ADRC can be found by delaying the controller output fed into the observer by $T_{\text{dead}}^{\text{ESO}}$, *i.e.*,
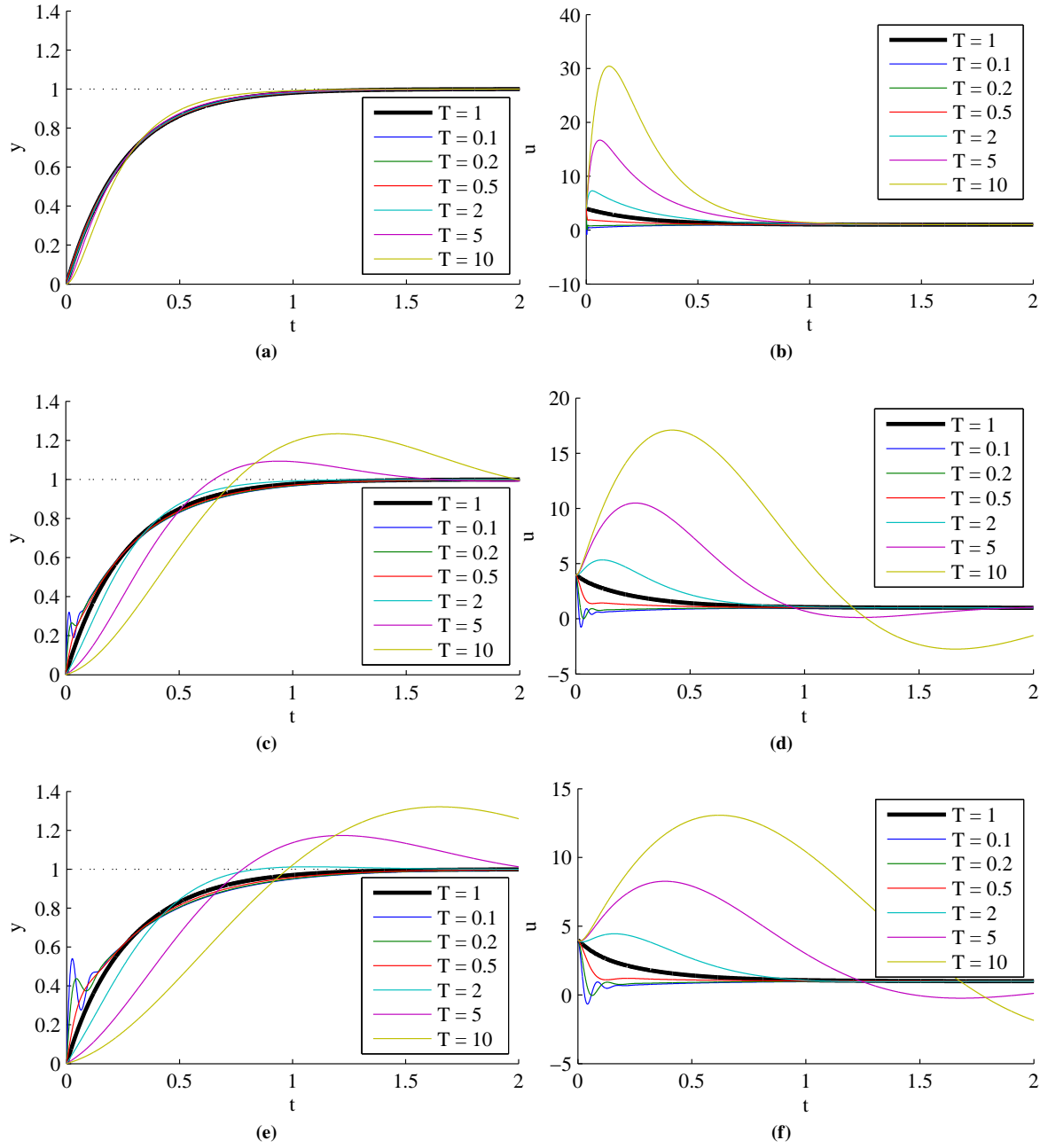
Figure 4. Experiment 3.1.2: effect of pole locations for the extended state observer (ESO) on a fixed first-order ADRC controlling first-order process with varying parameter, T. Nominal process parameters: $K = 1$, $T = 1$. ADRC parameters: $b_0 = \frac{K}{T} = 1$, $T_{\text{settle}} = 1$, s^{ESO} varying. (**a**) Closed loop step response, $s^{\text{ESO}} = 100 \cdot s^{\text{CL}}$; (**b**) Controller output u for (**a**); (**c**) Closed loop step response, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$; (**d**) Controller output u for (**c**); (**e**) Closed loop step response, $s^{\text{ESO}} = 5 \cdot s^{\text{CL}}$; (**f**) Controller output u for (**e**).

using $u(t - T_{\text{dead}}^{\text{ESO}})$ instead of $u(t)$ in Equation (4). In Figure 7c and (d), this approach was implemented using $T_{\text{dead}}^{\text{ESO}} = 0.05$. Clearly, the oscillations in the controller output are less prominent, even if $T_{\text{dead}}^{\text{ESO}}$ does not match the actual dead time.

3.1.5 Effect of Structural Uncertainties

In the experiments carried out so far in this section, it was assumed that our process could be reasonably well described by a first-order model. In practice, such a first-order model almost always results from neglecting higher-order dynamics, e.g. of the actuator. While it could already be seen that ADRC can handle variations of parameters
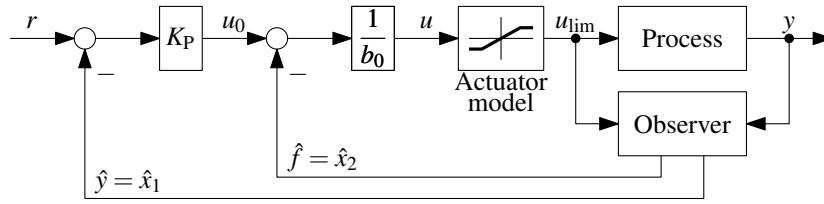
Figure 5. Control loop structure of first-order ADRC considering actuator saturation.
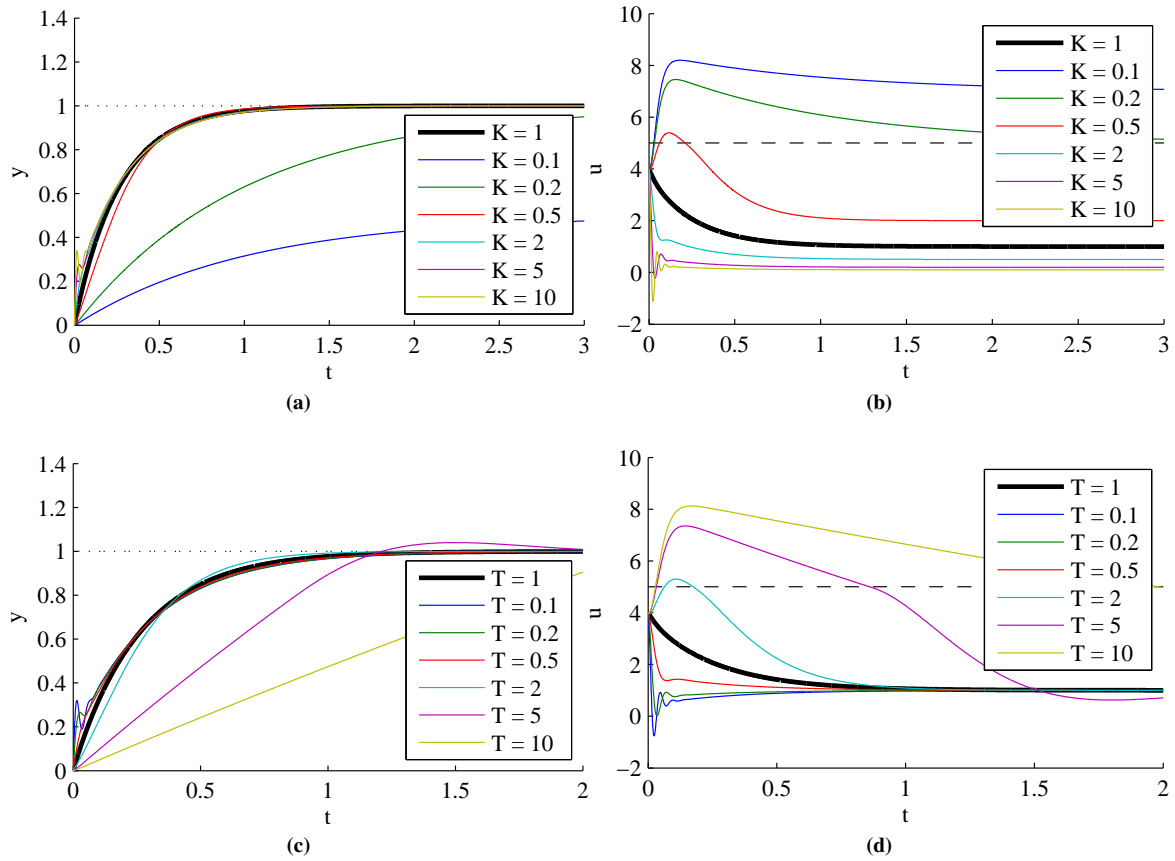

Figure 6. Experiment 3.1.3: effect of actuator saturation, $|u_{\text{lim}}| \leq 5$, on fixed a first-order ADRC controlling first-order process with varying parameters. Nominal process parameters: $K = 1, T = 1$. ADRC parameters: $b_0 = \frac{K}{T} = 1, T_{\text{settle}} = 1, s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$. On the right-hand side, the controller outputs are shown before the limitation (dashed line) takes effect. **(a)** Variation of K, closed loop step response; **(b)** Controller output u for **(a)**; **(c)** Variation of T, closed loop step response; **(d)** Controller output u for **(c)**.

very well, how does it behave if higher-order dynamics become unexpectedly visible in the process?

To demonstrate this behavior, a second pole was added to the process in the simulations from Figure 8, resulting in a second time constant, $T_2 \leq 0.1$, *i.e.*, up to 10% of the dominant time constant, T. One can see that some oscillations start to appear during the transient as the higher-order pole approaches the dominant pole. While these results are acceptable, further simulations showed that ADRC did not provide an advantage comparing to standard PI controllers as large as it does in the case of parameter robustness, *cf.* Section 3.1.6.

3.1.6 Comparison to PI Control

Given the ubiquity of PID-type controllers in industrial practice, how does the standard approach keep up against ADRC? To that end, we will repeat the experiment regarding sensitivity towards process parameter variations and compare the ADRC results to a PI controller.

For the first-order process, a PI controller is sufficient to achieve any desired first-order closed loop behavior.
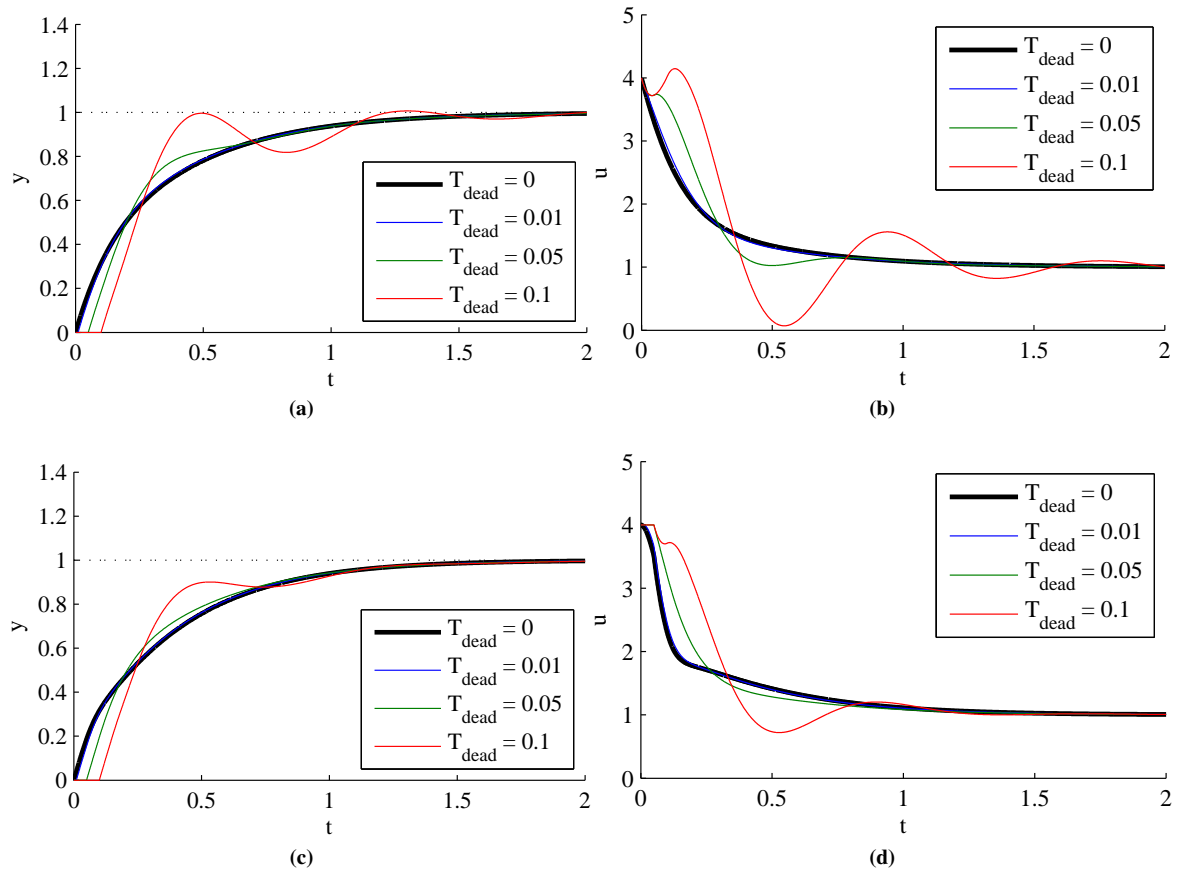
Figure 7. Experiment 3.1.4: effect of dead time, T_{dead}, on a first-order ADRC controlling a process with first-order dynamics ($K = 1$, $T = 1$) and unknown dead time. ADRC parameters: $b_0 = \frac{K}{T} = 1$, $T_{settle} = 1$, $s^{ESO} = 2 \cdot s^{CL}$. In (c) and (d), a fixed dead time, T_{dead}^{ESO}, was incorporated into the observer to improve the controller behavior. (a) Variation of T_{dead}, closed loop step response; (b) Controller output u for (a); (c) Closed loop step response, observer with $T_{dead}^{ESO} = 0.05$; (d) Controller output u for (c).

In order to obtain comparable results, the PI controller is designed for nearly identical closed loop dynamics as the ADRC by aiming for the same settling time and placing a zero on the pole of the first-order process:

$$C_{PI}(s) = K_P + \frac{K_I}{s} \quad \text{with} \quad K_I = \frac{3.85}{K \cdot T_{settle}} = 3.85 \quad \text{and} \quad K_P = K_I \cdot T = 3.85$$

The simulation results in Figure 9 clearly demonstrate the ability of the ADRC approach to keep the closed loop dynamics similar, even under major parameter variations. The PI controller delivers dynamics that vary heavily, as do the process parameters.

3.1.7 Disturbance Rejection of ADRC and PI

As a final experiment for the continuous-time first-order ADRC, we want to examine the disturbance rejection abilities by injecting an input disturbance into the process for both ADRC and the PI controller from Section 3.1.6.

On the left-hand side of Figure 10, a closed loop step response is shown. The input disturbance is effective during the period from $t = 2$ until $t = 4$. While both controllers were tuned for the same reaction on setpoint changes, the impact of the disturbance is compensated for much faster by ADRC compared to the PI controller. In classical control, one would need to tune the PI controller much more aggressively and add a setpoint filter or follow another 2DOF approach in order to obtain similar results [1].
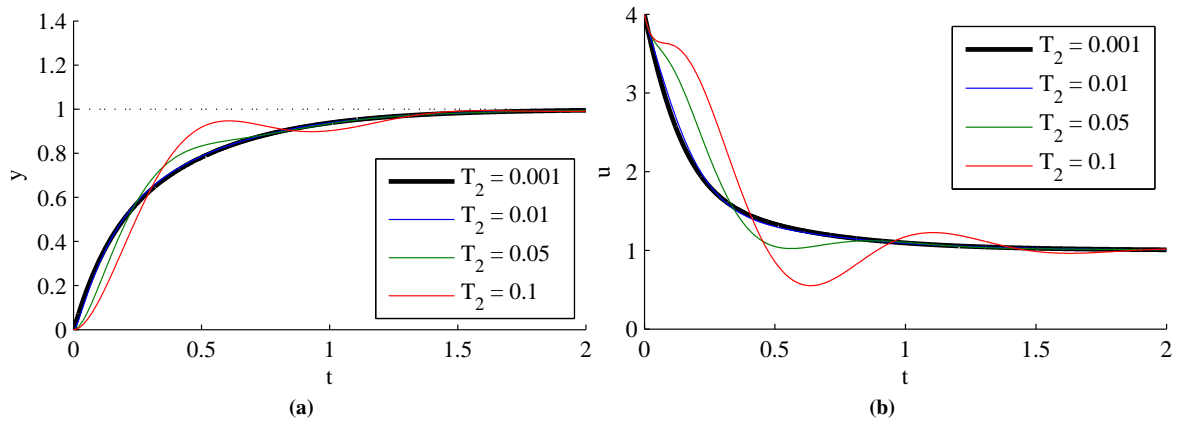
Figure 8. Experiment 3.1.5: effect of structural uncertainties on a first-order ADRC controlling a process with dominating first-order behavior ($K = 1, T = 1$) and higher-order dynamics caused by an unknown second pole at $s = -1/T_2$. ADRC parameters: $b_0 = \frac{K}{T} = 1, T_{\text{settle}} = 1, s^{\text{ESO}} = 2 \cdot s^{\text{CL}}$. (**a**) Variation of T_2, closed loop step response; (**b**) Controller output u for (**a**).
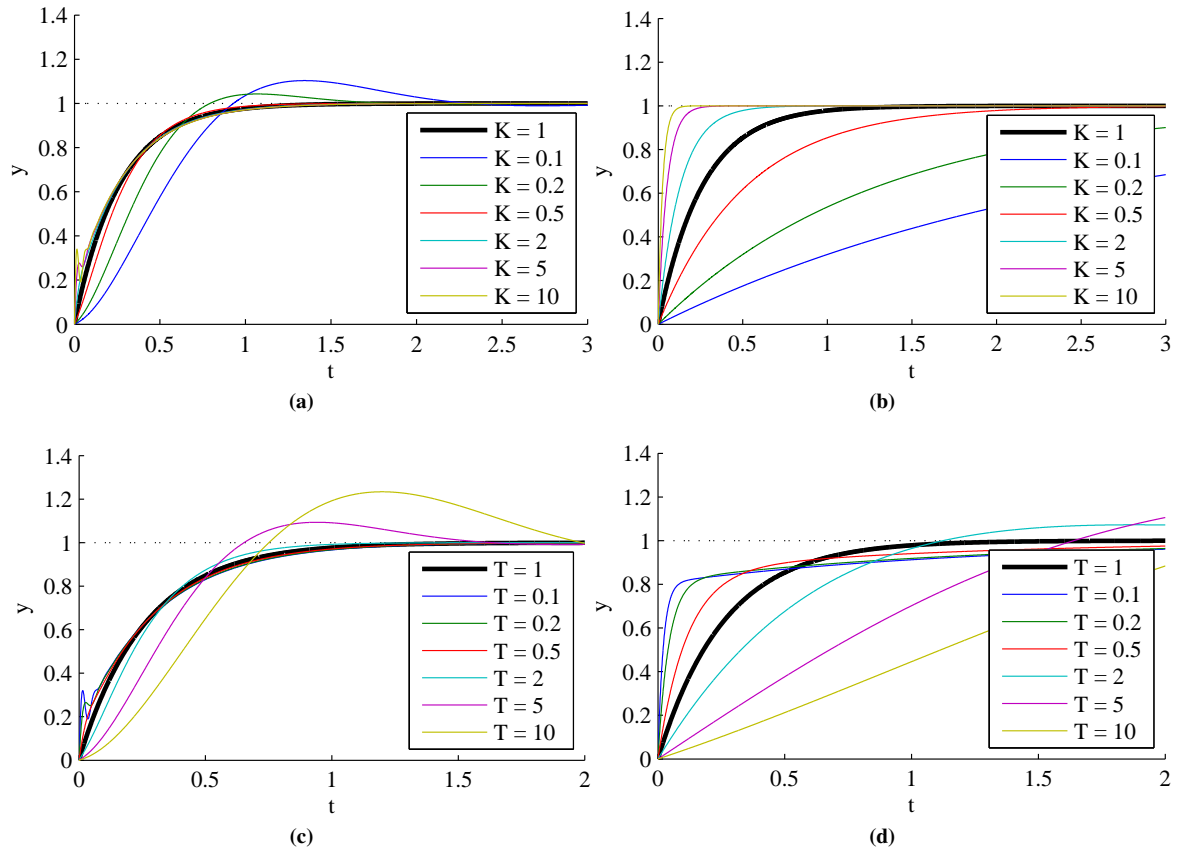

Figure 9. Experiment 3.1.6: comparison of first-order ADRC and PI controller faced with varying process parameters, K and T. Nominal process and ADRC parameters are as throughout Section 3.1. For each combination, the closed loop step response is shown. (**a**) Variation of K, ADRC; (**b**) Variation of K, PI; (**c**) Variation of T, ADRC; (**d**) Variation of T, PI.
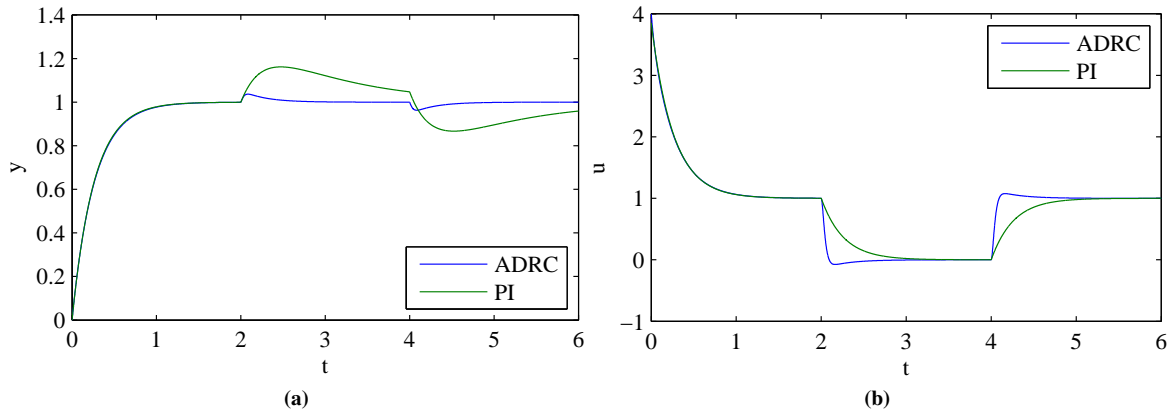
Figure 10. Experiment 3.1.7: comparison of disturbance rejection behavior (first-order ADRC *vs*. PI controller). Process and ADRC parameters are as throughout Section 3.1. Input disturbance, $d = 1$, was effective from $t = 2$ until $t = 4$. (**a**) Step response and reaction on disturbance; (**b**) Controller output u for (**a**).

3.2 Second-Order ADRC with a Second-Order Process

The second-order ADRC will be examined using a second-order process with the following nominal parameters:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{T^2 s + 2DTs + 1} \quad \text{with} \quad K = 1, \quad D = 1 \quad \text{and} \quad T = 1$$

Again, perfect knowledge of the process is assumed, $b_0 = \frac{K}{T^2} = 1$, but then b_0 is left unchanged throughout Section 3.2. The desired closed loop settling time is $T_{\text{settle}} = 5$. The parameters of the PD controller are, following Equation (14), set to $K_{\text{P}} = \left(\frac{6}{T_{\text{settle}}}\right)^2 = 1.44$ and $K_{\text{D}} = \frac{12}{T_{\text{settle}}} = 2.4$. Unless otherwise noted in individual experiments, the observer poles are chosen to be $s^{\text{ESO}}_{1/2/3} = s^{\text{ESO}} = 10 \cdot s^{\text{CL}} = 10 \cdot \left(-\frac{6}{T_{\text{settle}}}\right) = -12$. The respective values of the observer gains, $l_{1/2/3}$, are obtained via Equation (16). As in Section 3.1, throughout this section, noise-free control variables and ideal measurements will be assumed.

3.2.1 Sensitivity to Process Parameter Variations

For the second-order ADRC, sensitivity to variations of the process parameters will be examined firstly. Deviations from the original DC gain ($K = 1$) were made in a range of 10% to 500% by setting K to one of the values 0.1, 0.2, 0.5, 2 and 5. The damping factor, D, of the process was varied within a range of 10% to 1000% of the original value, $D = 1$, using settings 0.1, 0.2, 0.5, 2, 5 and 10. Finally, the time constant, T, of the second-order process was varied in a range of 50% to 350% of the original value, $T = 1$, using the values 0.5, 2, 3 and 3.5.

From the results in Figure 11, one can see that the closed loop behavior is almost not at all affected, even by relatively large changes of K and D. Only for very small values of K, the step response becomes slower; and for very large values of the damping, D, some overshoot becomes visible. Changes in the time constant, T, do, for larger values, increase overshoot and oscillations more than in the first-order case.

In order to fully appreciate the robustness towards parameter changes of the process, one has to compare these results against standard controllers, such as PID, as will be done in Section 3.2.6.

3.2.2 Effect of Observer Pole Locations

In the previous section, we saw that changes in the process time constant, T, affected the closed loop behavior stronger than changes in K and D. We will therefore demonstrate the influence of the observer poles on the sensitivity by repeating the experiments with varying T for different observer pole locations.

In the previous experiments, the observer poles were set ten times to the left of the closed loop poles in the s-plane, *i.e.*, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$. Here, both slower and faster observers will be examined, as well, by setting $s^{\text{ESO}} = 100 \cdot s^{\text{CL}}$ and $s^{\text{ESO}} = 5 \cdot s^{\text{CL}}$.

As visible in the simulation results in Figure 12, almost ideal behavior can—at least in theory—be obtained by placing the observer poles far enough left in the s-plane, *i.e.*, choosing large factors k in $s^{\text{ESO}} = k \cdot s^{\text{CL}}$. Of
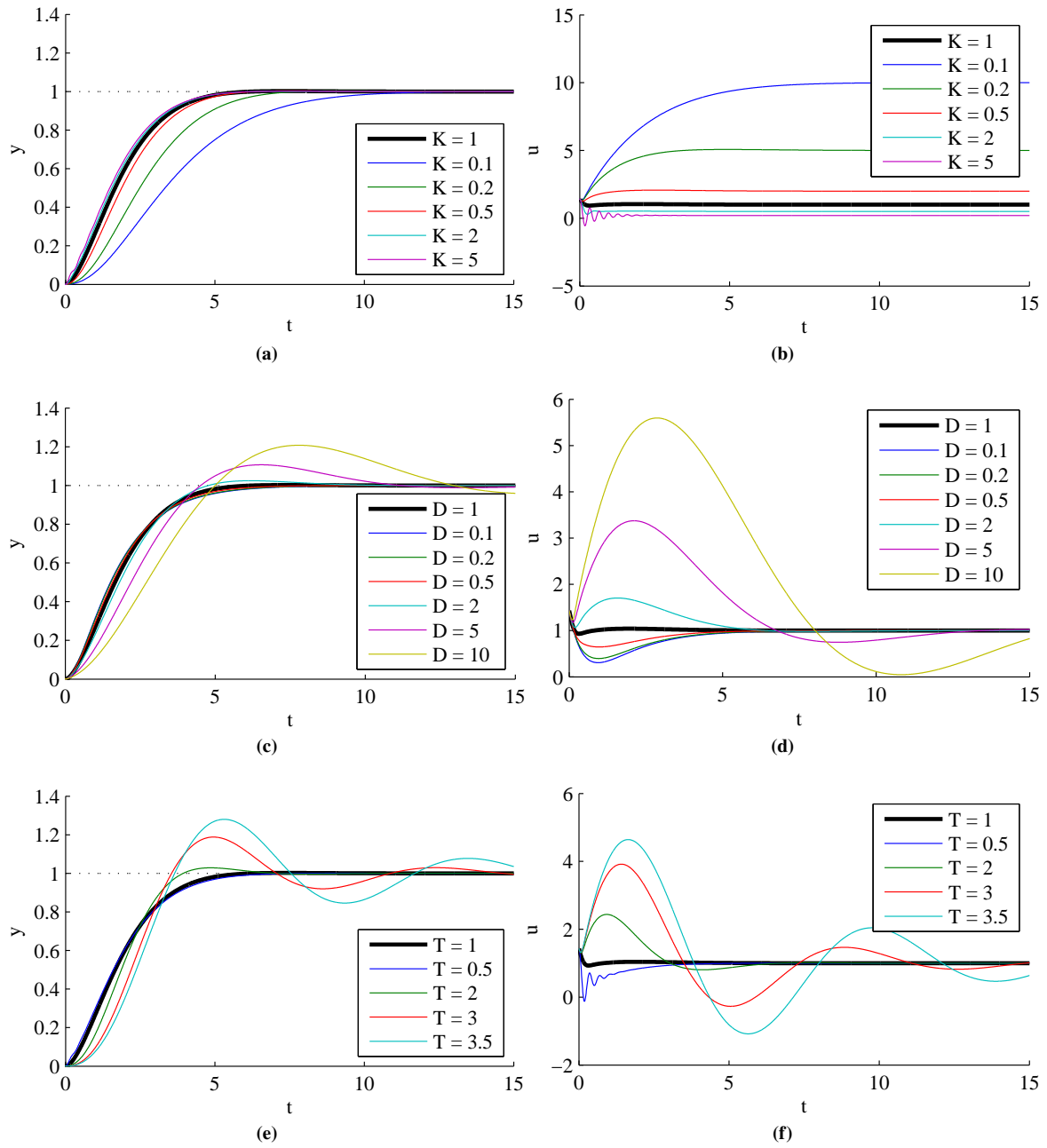
Figure 11. Experiment 3.2.1: a fixed second-order ADRC controlling second-order process with varying parameters. Nominal process parameters: $K = 1$, $D = 1$, $T = 1$. ADRC parameters: $b_0 = \frac{K}{T^2} = 1$, $T_{\text{settle}} = 5$, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$. (a) Variation of K, closed loop step response; (b) Controller output u for (a); (c) Variation of D, closed loop step response; (d) Controller output u for (c); (e) Variation of T, closed loop step response; (f) Controller output u for (e).

course, this comes at the price of the need for faster actuators and larger controller outputs. Furthermore—as we will see in Section 4.2—faster observers are more sensitive to measurement noise and will be limited by sample time restrictions in discrete time implementations.

3.2.3   Effect of Actuator Saturation

Analogously to Section 3.1.3, the experiments were carried out by extending the controller structure, such that the saturated controller output, $u_{\text{lim}}(t)$, was fed back to the observer instead of $u(t)$ (compare Figure 5 for the first-order case). The resulting step responses and controller outputs for varying process parameters can be found in Figure
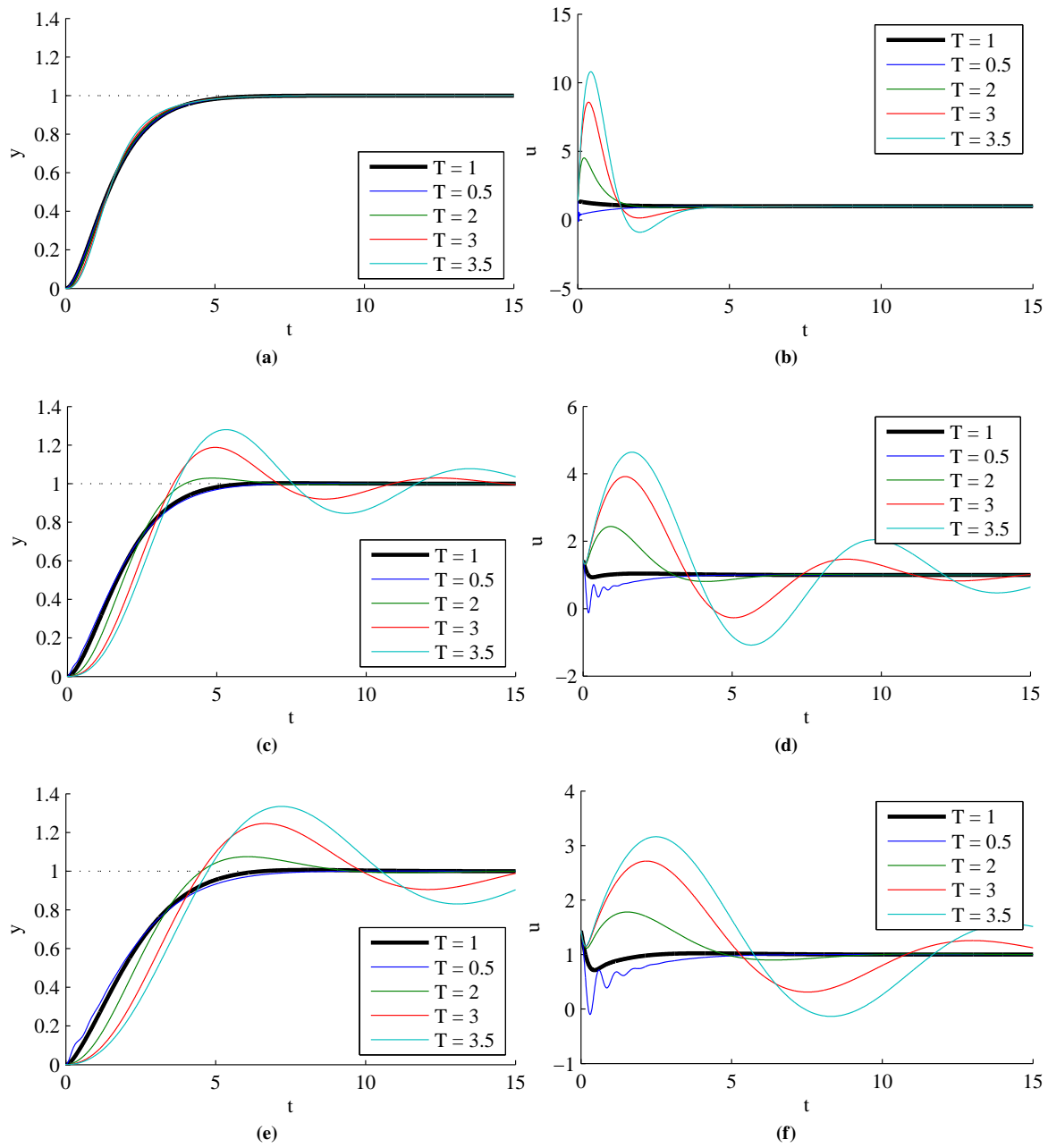
Figure 12. Experiment 3.2.2: effect of observer pole locations for a second-order ADRC controlling second-order processes with varying parameter, T. Nominal process parameters: $K = 1$, $D = 1$, $T = 1$. ADRC parameters: $b_0 = \frac{K}{T^2} = 1$, $T_{\text{settle}} = 5$, s^{ESO} varying. (**a**) Closed loop step response, $s^{\text{ESO}} = 100 \cdot s^{\text{CL}}$; (**b**) Controller output u for (**a**); (**c**) Closed loop step response, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$; (**d**) Controller output u for (**c**); (**e**) Closed loop step response, $s^{\text{ESO}} = 5 \cdot s^{\text{CL}}$; (**f**) Controller output u for (**e**).

13. As expected, lowering the DC gain, K, of the process forced the controller into saturation, such that the desired process output could not be reached anymore. It has to be stressed again that the controller output did not wind up, as visible in Figure 13.

When decelerating the process (increased damping D), the closed loop dynamics suffered when the controller output ran into saturation (increased settling time), but there were no additional oscillations after recovering from saturation, as known from classical PID control without anti-windup measures.
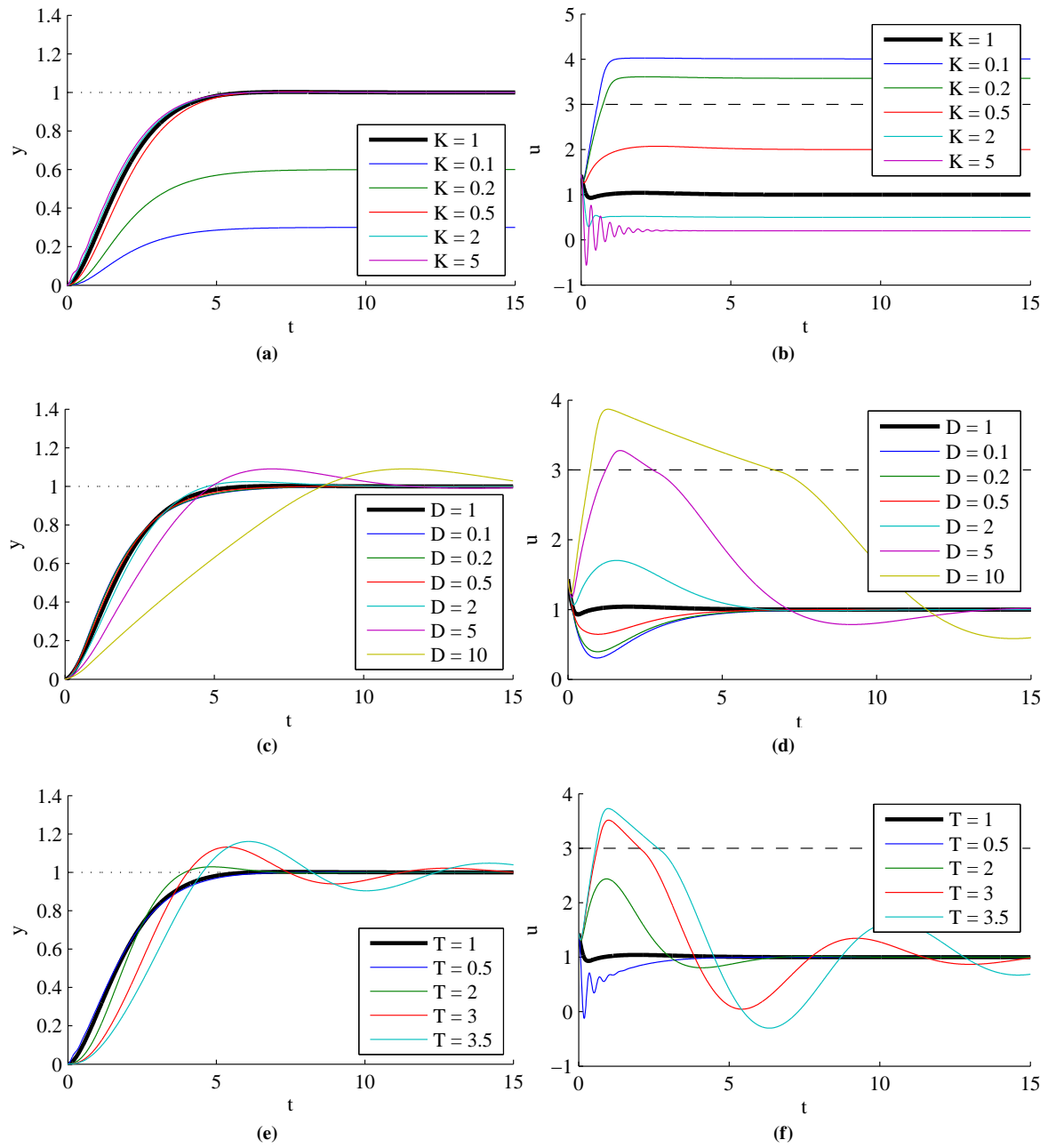
Figure 13. Experiment 3.2.3: effect of actuator saturation, $|u_{\text{lim}}| \leq 3$, on a fixed second-order ADRC controlling a second-order process with varying parameters. Nominal process parameters: $K = 1$, $D = 1$, $T = 1$. ADRC parameters: $b_0 = \frac{K}{T^2} = 1$, $T_{\text{settle}} = 5$, $s^{\text{ESO}} = 10 \cdot s^{\text{CL}}$. The controller outputs are shown before the limitation (dashed line) takes effect. (**a**) Variation of K, closed loop step response; (**b**) Controller output u for (**a**); (**c**) Variation of D, closed loop step response; (**d**) Controller output u for (**c**); (**e**) Variation of T, closed loop step response; (**f**) Controller output u for (**e**).

3.2.4 Effect of Dead Time

Similar to Section 3.1.4, the second-order ADRC was confronted with an unknown dead time in the process with $T_{\text{dead}} \leq 0.3$. One can see from Figure 14a that the controller and observer work very well together, such that the output is hardly being affected by the dead time, which is also an improvement compared to the first-order case in Section 3.1.4. In the controller output in Figure 14b, however, oscillations become increasingly visible with larger values of T_{dead}. Again, this situation can be improved by delaying the input of the observer, *i.e.*, using $u(t - T_{\text{dead}}^{\text{ESO}})$ instead of $u(t)$ in Equation (12). For a fixed value, $T_{\text{dead}}^{\text{ESO}} = 0.1$, the controller behavior is shown in Figure 14c and
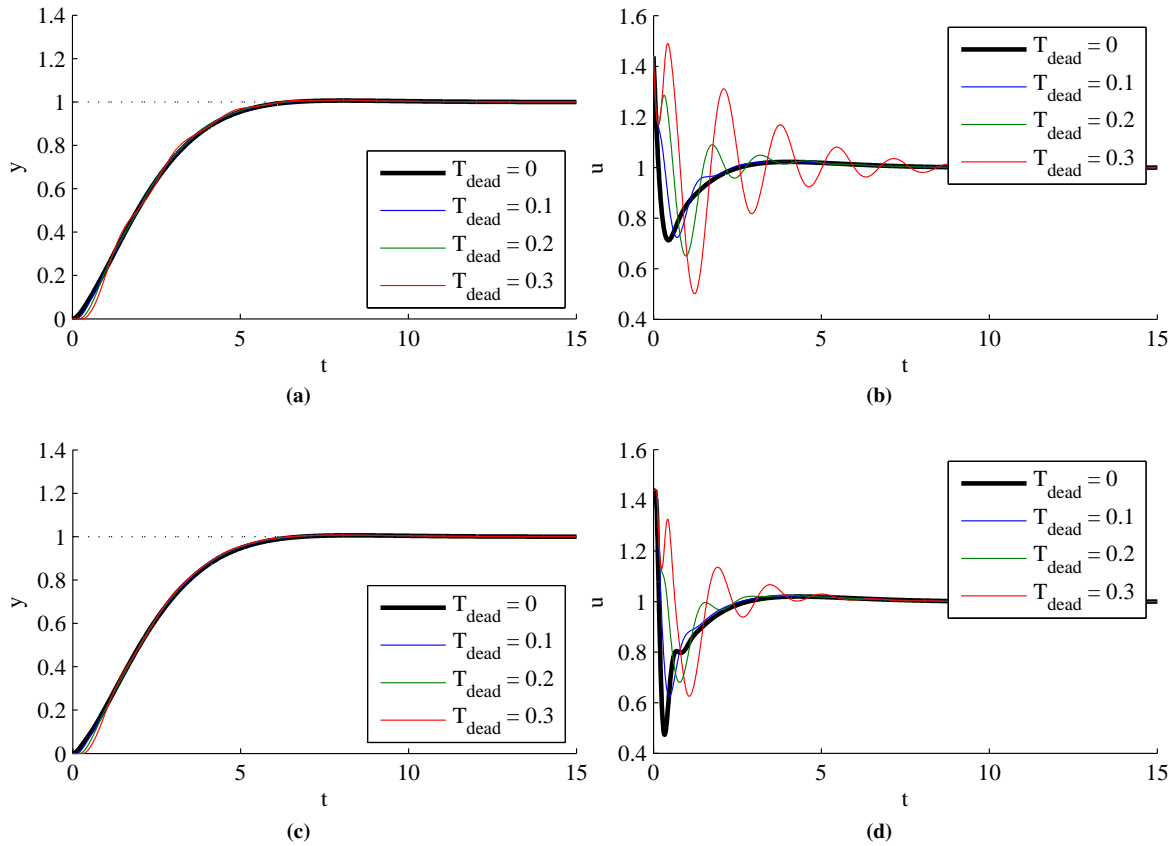
Figure 14. Experiment 3.2.4: effect of dead time, T_{dead}, on a second-order ADRC controlling a process with second-order dynamics ($K = 1$, $D = 1$, $T = 1$) and unknown dead time. ADRC parameters: $b_0 = \frac{K}{T^2} = 1$, $T_{\text{settle}} = 5$, $s^{\text{ESO}} = 5 \cdot s^{\text{CL}}$. In (**c**) and (**d**), a fixed dead time, $T_{\text{dead}}^{\text{ESO}}$, was incorporated into the observer to improve the controller behavior. (**a**) Variation of T_{dead}, closed loop step response; (**b**) Controller output u for (**a**); (**c**) Closed loop step response, observer with $T_{\text{dead}}^{\text{ESO}} = 0.1$; (**d**) Controller output u for (**a**).

(d), where the controller oscillations are already significantly reduced, even if $T_{\text{dead}}^{\text{ESO}}$ does not match the actual dead time.

3.2.5 Effect of Structural Uncertainties

Besides unknown dead times, the controller may be faced with higher-order dynamics in the process. Therefore, the effect of an unknown third pole in the process was examined in Figure 15. The time constant, T_3, of the third pole was varied within $0.001 \leq T_3 \leq 1$, *i.e.*, the third pole was identical to the two known poles of the plant in the extreme case. In contrast to the first-order ADRC in Section 3.1.5, the second-order ADRC proved to have very good robustness against an unknown higher-order dynamics, even in the more challenging cases of Figure 15.

3.2.6 Comparison to PI and PID Control

To view and assess the abilities of the second-order ADRC from a perspective of classical control, we will now employ standard PI and PID controllers and—after fixing their parameters—expose them to the same process parameter variations, as done in Section 3.2.1. To ensure comparability, all controllers are being designed for the same closed loop dynamics (settling time $T_{\text{settle}} = 5$, no overshoot) using the nominal process parameters, $K = 1$, $D = 1$, $T = 1$.

The PI controller is given and parameterized as follows:

$$C_{\text{PI}}(s) = K_{\text{P}} + \frac{K_{\text{I}}}{s} \quad \text{with} \quad K_{\text{I}} = \frac{2.55}{K \cdot T_{\text{settle}}} = 0.51 \quad \text{and} \quad K_{\text{P}} = K_{\text{I}} \cdot 1.5 \cdot D \cdot T = 0.765$$
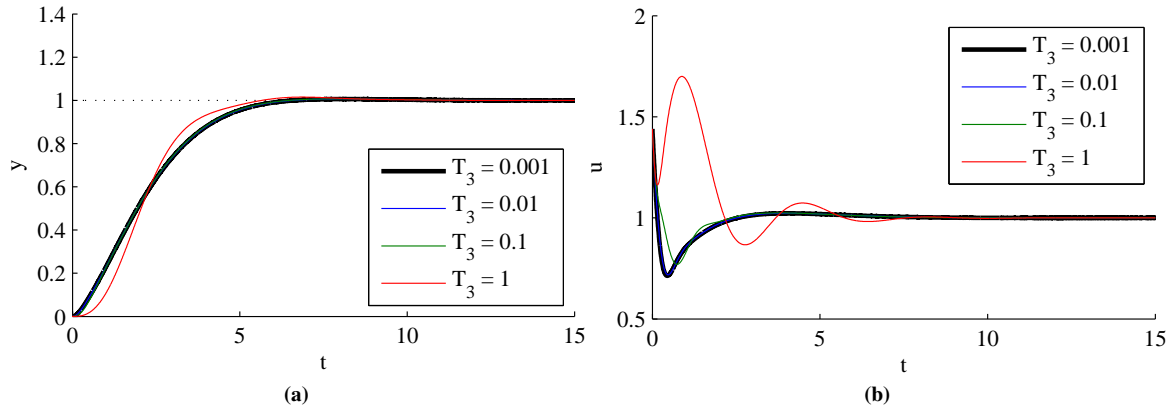
Figure 15. Experiment 3.2.5: effect of structural uncertainties on a second-order ADRC controlling a process with dominating second-order behavior ($K = 1$, $D = 1$, $T = 1$) and higher-order dynamics caused by an unknown third pole at $s = -1/T_3$. ADRC parameters: $b_0 = \frac{K}{T^2} = 1$, $T_{\text{settle}} = 5$, $s^{\text{ESO}} = 5 \cdot s^{\text{CL}}$. (**a**) Variation of T_3, closed loop step response; (**b**) Controller output u for (**a**).
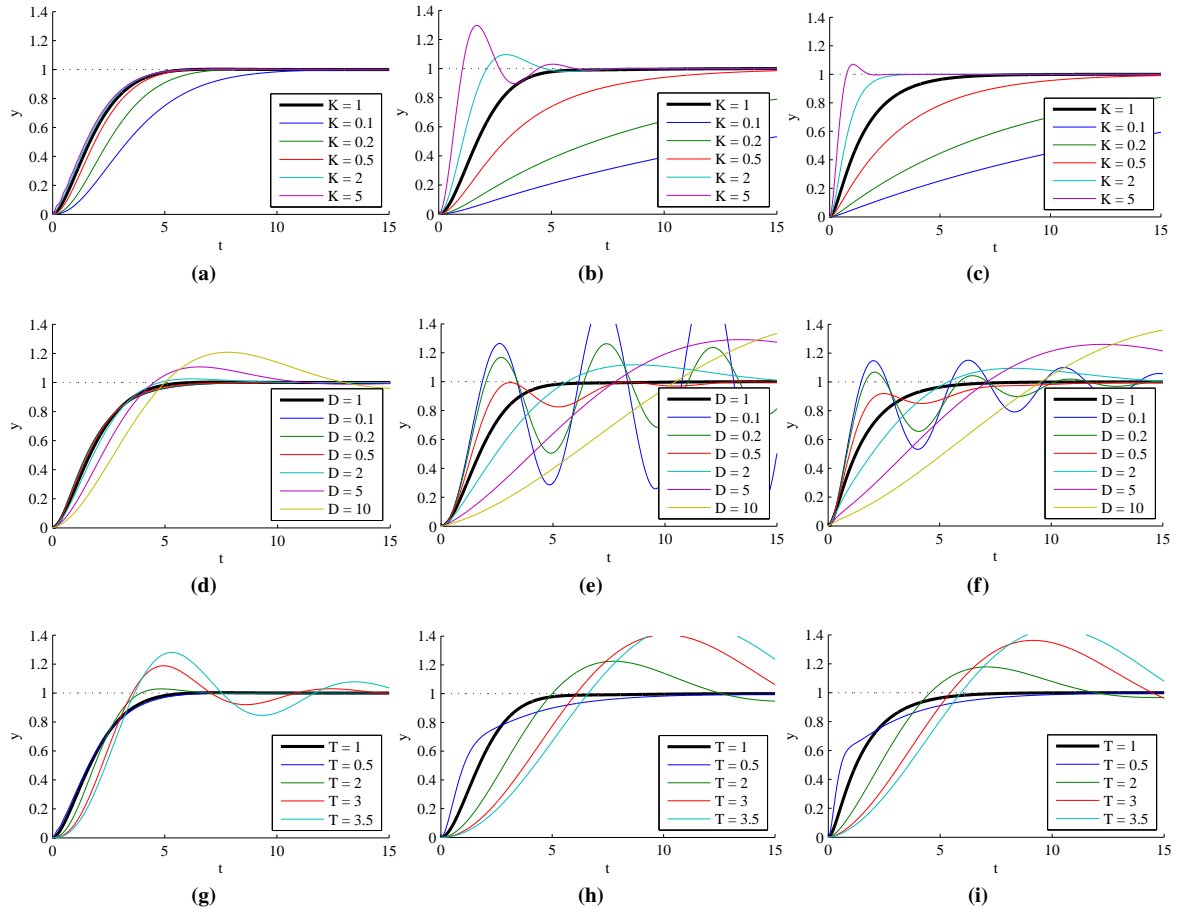
Figure 16. Experiment 3.2.6: comparison of second-order ADRC, PI and PID controllers faced with varying process parameters, K, D and T. Nominal process and ADRC parameters are as throughout Section 3.2. For each combination, the closed loop step response is shown. (**a**) Variation of K, ADRC; (**b**) Variation of K, PI; (**c**) Variation of K, PID; (**d**) Variation of D, ADRC; (**e**) Variation of D, PI; (**f**) Variation of D, PID; (**g**) Variation of T, ADRC; (**h**) Variation of T, PI; (**i**) Variation of T, PID.

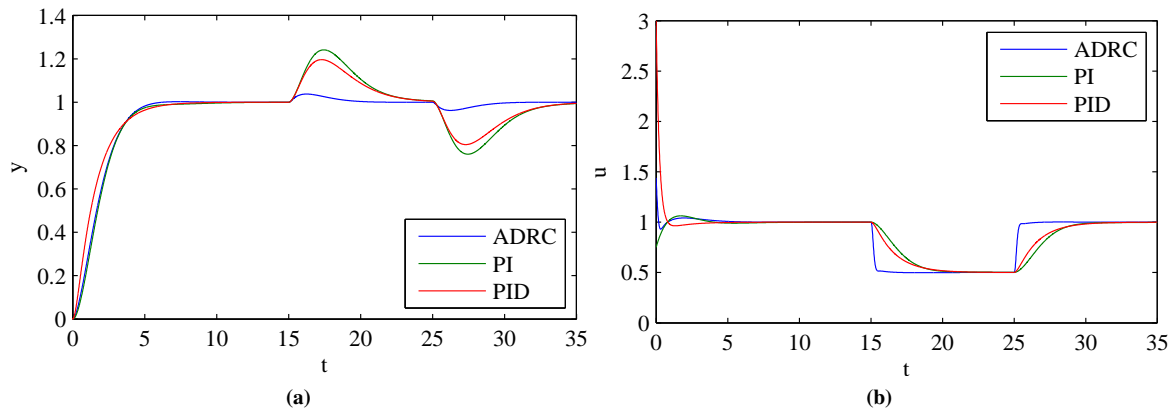**Figure 17.** Experiment 3.2.7: comparison of disturbance rejection behavior (second-order ADRC *vs.* PI *vs.* PID controller). Process and ADRC parameters are as throughout Section 3.2. Input disturbance, $d = 0.5$, effective from $t = 15$ until $t = 25$. (**a**) Step response and reaction on disturbance; (**b**) Controller output u for (**a**).

For the PID controller, we will employ a PIDT$_1$-type controller in two-pole-two-zero form, designed such that the zeros of the controller cancel the process poles. The controller gain is chosen to match the closed loop dynamics to the PI and ADRC case:

$$C_{\text{PID}}(s) = K_{\text{I}} \cdot \frac{(1 + T_{\text{Z1}} \cdot s)(1 + T_{\text{Z2}} \cdot s)}{s \cdot (1 + T_1 \cdot s)} \quad \text{with} \quad K_{\text{I}} = \frac{3}{K \cdot T_{\text{settle}}} = 0.6, \quad T_{\text{Z1/2}} = T = 1 \quad \text{and} \quad T_1 = 0.2$$

The simulation results are compiled in Figure 16, where each column represents one of the three controller types and each row, a different process parameter being varied. To summarize, it has to be stated that in each case, the ADRC approach surpasses the results of PI and PID control by a large margin with respect to sensitivity towards parameter variations, with a slight advantage of PID compared to PI control.

3.2.7   Disturbance Rejection of ADRC, PI and PID

As our final experiment for the second-order case, we will evaluate the disturbance rejection capabilities of ADRC, PI and PID control using the controller settings from the previous section. All of them are tuned for the same closed loop response.

For each control loop, an input disturbance, $d = 0.5$, is applied during a ten-second interval after reaching steady state; compare Figure 17. One can easily recognize that ADRC compensates the disturbance much faster, such that its effect on the control variable remains very low. While both PI and PID controllers could also be tuned more aggressively for a similar disturbance rejection behavior, one would have to employ additional measures, such as setpoint filters, to retain a non-oscillating reference tracking behavior.

4   Discrete Time ADRC

Practical implementations of a controller with a state observer, such as the ADRC approach, will most likely be done in discrete time form, e.g. employing a microcontroller. Since the actual control law for linear ADRC is based on proportional state feedback, a discrete time version can already be obtained by only discretizing the extended state observer, which will be done in Section 4.1. The quasi-continuous approach will be valid only for sufficiently fast sampling, of course. Otherwise, the state feedback should be designed explicitly for a discretized process model incorporating sampling delays.

In Section 4.2, simulative experiments will be carried out in order to visually assess the influence of the discretisation process and measurement noise on the control loop performance.

4.1   Discretisation of the State Observer

For a system without a feed-through term, the standard approach for a discrete-time observer is:

$$\hat{\boldsymbol{x}}(k + 1) = \boldsymbol{A}_{\text{d}} \cdot \hat{\boldsymbol{x}}(k) + \boldsymbol{B}_{\text{d}} \cdot \boldsymbol{u}(k) + \boldsymbol{L}_{\text{p}} \cdot (y(k) - \boldsymbol{C}_{\text{d}} \cdot \hat{\boldsymbol{x}}(k)) \tag{22}$$

If we look at the equation for the estimation error of Equation (22), we can see that—just as in the continuous case—the dynamics of the error decay are determined by the matrix $(A_\mathrm{d} - L_\mathrm{p} \cdot C_\mathrm{d})$:

$$e(k + 1) = x(k + 1) - \hat{x}(k + 1) = (A_\mathrm{d} - L_\mathrm{p} \cdot C_\mathrm{d}) \cdot (x(k) - \hat{x}(k))$$

Since the observer gains in L_p influence the pole placement for the matrix $(A_\mathrm{d} - L_\mathrm{p} \cdot C_\mathrm{d})$, they can be chosen such that the estimation settles within a desired time.

Here, A_d, B_d and C_d refer to time-discrete versions of the respective matrices in the state space process models Equations (3) and (11) obtained by a discretisation method. Since there is no matrix D_d in the observer equations, the discretisation of the model must deliver $D_\mathrm{d} = 0$, for example, via zero order hold (ZOH) sampling. This observer approach is also being called "prediction observer", since the current measurement, $y(k)$, will be used as a correction of the estimate only in the subsequent time step, $\hat{x}(k + 1)$.

In order to reduce unnecessary time delays (which may even destabilize the control loop), it is advisable [10] to employ a different observer approach called "filtered observer" or "current observer" [5]. The underlying idea is—similar to Kalman filtering—to split the update equation in two steps, namely a prediction step to predict $\hat{x}(k|k - 1)$ based on measurements of the previous time step, $k - 1$, and a correction step to obtain the final estimate, $\hat{x}(k|k)$, incorporating the most recent measurement, $y(k)$:

$$\hat{x}(k|k - 1) = A_\mathrm{d} \cdot \hat{x}(k - 1|k - 1) + B_\mathrm{d} \cdot u(k - 1) \quad \text{(prediction)}$$

$$\hat{x}(k|k) = \hat{x}(k|k - 1) + L_\mathrm{c} \cdot (y(k) - C_\mathrm{d} \cdot \hat{x}(k|k - 1)) \quad \text{(correction)}$$

If we put the prediction into the correction equation and abbreviate $\hat{x}(k|k) = \hat{x}(k)$, we obtain one update equation for the observer:

$$\hat{x}(k) = (A_\mathrm{d} - L_\mathrm{c} \cdot C_\mathrm{d} \cdot A_\mathrm{d}) \cdot \hat{x}(k - 1) + (B_\mathrm{d} - L_\mathrm{c} \cdot C_\mathrm{d} \cdot B_\mathrm{d}) \cdot u(k - 1) + L_\mathrm{c} \cdot y(k) \tag{23}$$

From the estimation error, we can see that, in contrast to the prediction observer, the error dynamics are determined by the matrix $(A_\mathrm{d} - L_\mathrm{c} \cdot C_\mathrm{d} \cdot A_\mathrm{d})$:

$$e(k + 1) = x(k + 1) - \hat{x}(k + 1) = (A_\mathrm{d} - L_\mathrm{c} \cdot C_\mathrm{d} \cdot A_\mathrm{d}) \cdot (x(k) - \hat{x}(k))$$

When computing the observer gains in L_c, this has to be taken into account, *i.e.*, one must choose L_c, such that the eigenvalues of $(A_\mathrm{d} - L_\mathrm{c} \cdot C_\mathrm{d} \cdot A_\mathrm{d})$ match the desired observer pole locations.

The discrete time versions of the matrices A, B and C from the the state space process models, which are necessary for the observer equations, can be obtained by ZOH discretisation [5]:

$$A_\mathrm{d} = I + \sum_{i=1}^{\infty} \frac{A^i \cdot T_\text{sample}^i}{i!}, \quad B_\mathrm{d} = \left(\sum_{i=1}^{\infty} \frac{A^{i-1} \cdot T_\text{sample}^i}{i!}\right) \cdot B, \quad C_\mathrm{d} = C, \quad D_\mathrm{d} = D \tag{24}$$

For the first-order process in Equation (3), A and B are being discretized as follows, while $C_\mathrm{d} = C = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $D_\mathrm{d} = D = 0$ remain unchanged:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} b_0 \\ 0 \end{pmatrix} \quad \text{gives} \quad A_\mathrm{d} = \begin{pmatrix} 1 & T_\text{sample} \\ 0 & 1 \end{pmatrix}, \quad B_\mathrm{d} = \begin{pmatrix} b_0 \cdot T_\text{sample} \\ 0 \end{pmatrix}$$

This can be computed via Equation (24) easily, since $A^i = 0$ for $i \geq 2$. Following the same procedure for the second-order case in Equation (11), $C_\mathrm{d} = C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ and $D_\mathrm{d} = D = 0$ remain unchanged, and one obtains for A_d and B_d (since $A^i = 0$ for $i \geq 3$):

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix} \quad \text{gives} \quad A_\mathrm{d} = \begin{pmatrix} 1 & T_\text{sample} & T_\text{sample}^2/2 \\ 0 & 1 & T_\text{sample} \\ 0 & 0 & 1 \end{pmatrix}, \quad B_\mathrm{d} = \begin{pmatrix} b_0 \cdot T_\text{sample}^2/2 \\ b_0 \cdot T_\text{sample} \\ 0 \end{pmatrix}$$

Now, one can compute the observer gain, $L_\mathrm{c} = \begin{pmatrix} l_1 & l_2 \end{pmatrix}^\mathrm{T}$ (first-order case) or $L_\mathrm{c} = \begin{pmatrix} l_1 & l_2 & l_3 \end{pmatrix}^\mathrm{T}$ (for the second-order ADRC) to obtain the desired observer dynamics. The desired pole locations can, in a first step, be formulated in the s-plane, as in Sections 2.1 and 2.2, and then be mapped to the z-plane: $z^\mathrm{ESO} = e^{s^\mathrm{ESO} \cdot T_\text{sample}}$. The

computation of $\boldsymbol{L}_c$ may either be done numerically or analytically. To demonstrate this, we will derive the equations for the first-order case with one common pole location, z^{ESO}, for both poles:

$$\det\left(z\boldsymbol{I} - (\boldsymbol{A}_d - \boldsymbol{L}_c \cdot \boldsymbol{C}_d \cdot \boldsymbol{A}_d)\right) \overset{!}{=} \left(z - z^{ESO}\right)^2$$

$$\det\begin{pmatrix} z + l_1 - 1 & l_1 \cdot T_{sample} - T_{sample} \\ l_2 & z + l_2 \cdot T_{sample} - 1 \end{pmatrix} \overset{!}{=} z^2 - 2 \cdot z^{ESO} \cdot z - \left(z^{ESO}\right)^2$$

$$z^2 + \left(l_1 + l_2 \cdot T_{sample} - 2\right) \cdot z + (1 - l_1) \overset{!}{=} z^2 - 2 \cdot z^{ESO} \cdot z - \left(z^{ESO}\right)^2$$

By comparing the coefficients, one obtains for l_1 and l_2 (solutions for higher-order observers can be found in [10]):

$$l_1 = 1 - \left(z^{ESO}\right)^2, \quad l_2 = \frac{1}{T_{sample}} \cdot \left(1 - \left(z^{ESO}\right)^2\right) \tag{25}$$

As in the continuous time case in Section 2, the implementation and design of a discrete time linear ADRC can be summarized in four steps, where the first-order and second-order case are distinguished by (a) and (b):

1. *Modeling:*

   (a) For a process with dominating first-order behavior, $P(s) = \dfrac{K}{Ts + 1}$, one should provide an estimate $b_0 \approx \frac{K}{T}$.

   (b) For a second-order process, $P(s) = \dfrac{K}{T^2 s^2 + 2DTs + 1}$, an approximate value $b_0 \approx \frac{K}{T^2}$ is sufficient.

2. *Control structure:*

   Implement a discrete-time observer with two state variables, $\hat{\boldsymbol{x}}(k) = \begin{pmatrix} \hat{x}_1(k) & \hat{x}_2(k) \end{pmatrix}^T$ (for the first-order case) or three state variables $\hat{\boldsymbol{x}}(k) = \begin{pmatrix} \hat{x}_1(k) & \hat{x}_2(k) & \hat{x}_3(k) \end{pmatrix}^T$ (for the second-order case), as follows:

   $$\hat{\boldsymbol{x}}(k) = (\boldsymbol{A}_d - \boldsymbol{L}_c \cdot \boldsymbol{C}_d \cdot \boldsymbol{A}_d) \cdot \hat{\boldsymbol{x}}(k - 1) + (\boldsymbol{B}_d - \boldsymbol{L}_c \cdot \boldsymbol{C}_d \cdot \boldsymbol{B}_d) \cdot u(k - 1) + \boldsymbol{L}_c \cdot y(k)$$

   Using the estimated state variables, implement a proportional state feedback controller:

   (a) $u(k) = \dfrac{K_P \cdot (r(k) - \hat{y}(k)) - \hat{f}(k)}{b_0} = \dfrac{K_P \cdot (r(k) - \hat{x}_1(k)) - \hat{x}_2(k)}{b_0}$

   (b) $u(k) = \dfrac{\left(K_P \cdot (r(k) - \hat{y}(k)) - K_D \cdot \hat{\dot{y}}(k)\right) - \hat{f}(k)}{b_0} = \dfrac{\left(K_P \cdot (r(k) - \hat{x}_1(k)) - K_D \cdot \hat{x}_2(k)\right) - \hat{x}_3(k)}{b_0}$

3. *Closed loop dynamics:*

   Choose the controller parameters, e.g. via pole placement according to a desired settling time:

   (a) $K_P = -s^{CL} \quad \text{with} \quad s^{CL} \approx -\dfrac{4}{T_{settle}}$

   (b) $K_P = \left(s^{CL}\right)^2, \quad K_D = -2 \cdot s^{CL} \quad \text{with} \quad s^{CL} \approx -\dfrac{6}{T_{settle}}$

4. *Observer dynamics:*

   Place the desired observer poles far enough left of the closed loop poles in the s-plane, e.g. by a common location, $s^{ESO} = k_{ESO} \cdot s^{CL} \approx (3 \dots 10) \cdot s^{CL}$. Map to the z-plane via $z^{ESO} = e^{s^{ESO} \cdot T_{sample}}$ and compute the necessary observer gains:

   (a) $l_1 = 1 - \left(z^{ESO}\right)^2, \quad l_2 = \frac{1}{T_{sample}} \cdot \left(1 - z^{ESO}\right)^2$

   (b) $l_1 = 1 - \left(z^{ESO}\right)^3, \quad l_2 = \frac{3}{2 \cdot T_{sample}} \cdot \left(1 - z^{ESO}\right)^2 \cdot \left(1 + z^{ESO}\right), \quad l_3 = \frac{1}{T_{sample}^2} \cdot \left(1 - z^{ESO}\right)^3$
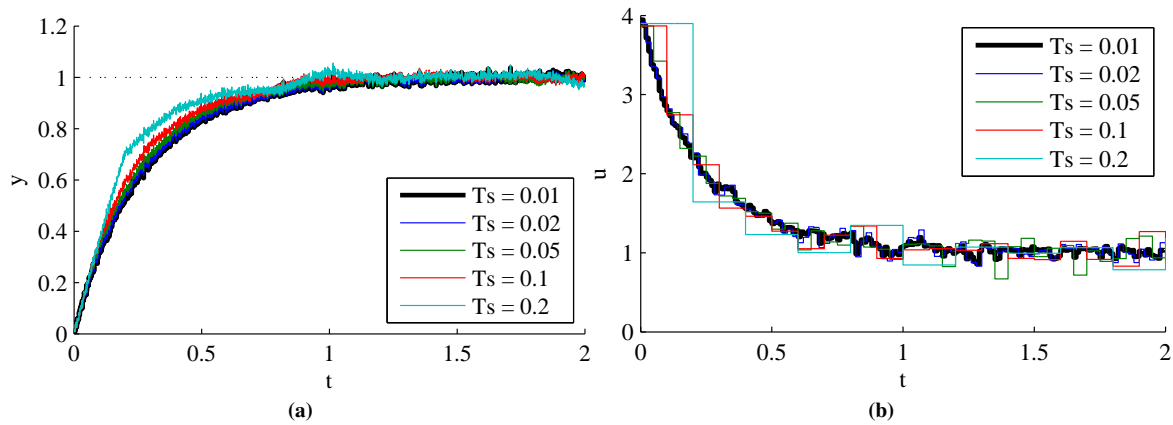
Figure 18. Experiment 4.2.1: influence of different sample time settings on discrete-time ADRC. Measurement noise variance: $\sigma^2_{\text{noise}} = 0.0001$. Observer pole placement: $k_{\text{ESO}} = 5$. (**a**) Variation of T_{sample}, closed loop step response; (**b**) Controller output u for (**a**).

4.2 Simulative Experiments

As in the continuous-time experiments in Section 3.1, a first-order process will be examined in this section:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{Ts + 1} \quad \text{with} \quad K = 1 \quad \text{and} \quad T = 1$$

Unless otherwise noted in individual experiments, the ADRC will be designed assuming perfect knowledge, $b_0 = K/T = 1$, with $k_{\text{ESO}} = 5$, and a desired 2% settling time, $T_{\text{settle}} = 1$. The discretisation will be based on a sampling time $T_{\text{sample}} = 0.01$. Gaussian measurement noise will be added with a variance $\sigma^2_{\text{noise}} = 0.0001$.

4.2.1 Effect of Sample Time

When choosing the sampling time for a discrete-time implementation of a controller with an observer, one has to consider not only the process dynamics, but also the dynamics of the observer. If, on the other hand, restrictions to sample times are present, the dynamics of the observer will be limited. In the case of ADRC, this would mean that the desired behavior of the process may not be achieved under all circumstances.

In Figure 18, simulations were performed using sample times for ADRC ranging from $T_{\text{sample}} = 0.01$ to $T_{\text{sample}} = 0.20$. As expected, oscillations in the controller output are increasingly visible for large sample times. Consequently, the closed loop dynamics differ from the desired first-order behavior as the sampling interval becomes too coarse.

4.2.2 Effect of Measurement Noise

While state feedback controllers based on observers rather than direct measurements are less susceptible to measurement noise, oscillations in the controller output will still occur with higher noise levels. In the simulations presented in Figure 19, normally distributed noise with increasing variance ranging from $\sigma^2_{\text{noise}} = 0$ to $\sigma^2_{\text{noise}} = 0.001$ was added to the process output. The results correspond to these expectations. The effect of measurement noise on oscillations in the controller output, however, may be mitigated by designing an observer with slower dynamics, which will be demonstrated in the following experiment.

4.2.3 Effect of Observer Pole Locations

In the simulations with varying observer dynamics presented in Figure 20, the typical trade-off between fast setpoint tracking and noise rejection becomes visible. If the observer poles are placed far enough left of the closed loop poles in the s-plane via k_{ESO}, the closed loop dynamics can adopt the desired first-order behavior, but the control variable becomes more sensitive to measurement noise and increasingly exhibits unwanted oscillations. On the other hand, placing observer poles near the closed loop poles provides good measures to suppress the effect of measurement noise on the control action, but the closed loop dynamics will suffer, especially under process parameter changes, *cf*. Sections 3.1.2 and 3.2.2.
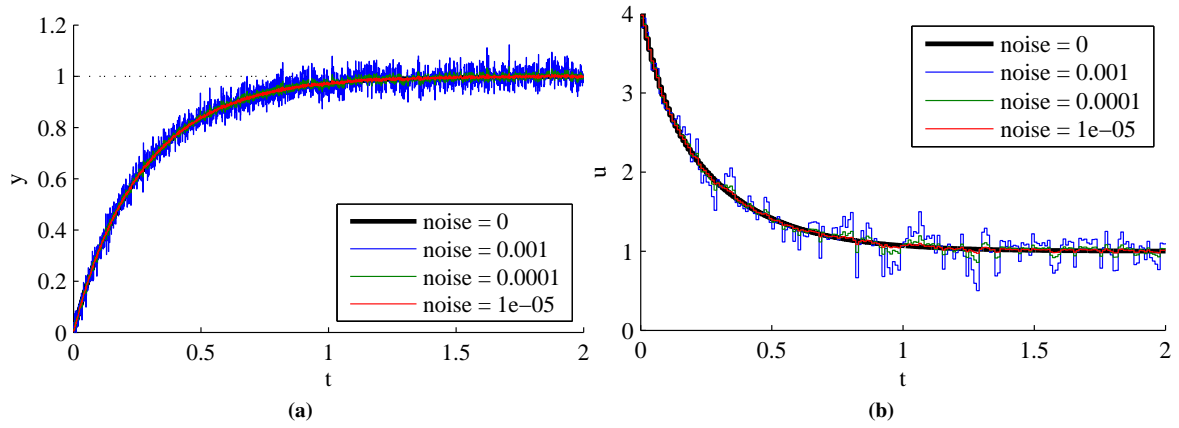
Figure 19. Experiment 4.2.2: influence of different levels of measurement noise on discrete-time ADRC. Sample time: $T_{\text{sample}} = 0.01$. Observer pole placement: $k_{\text{ESO}} = 5$. (a) Variation of σ^2_{noise}, closed loop step response; (b) Controller output u for (a).
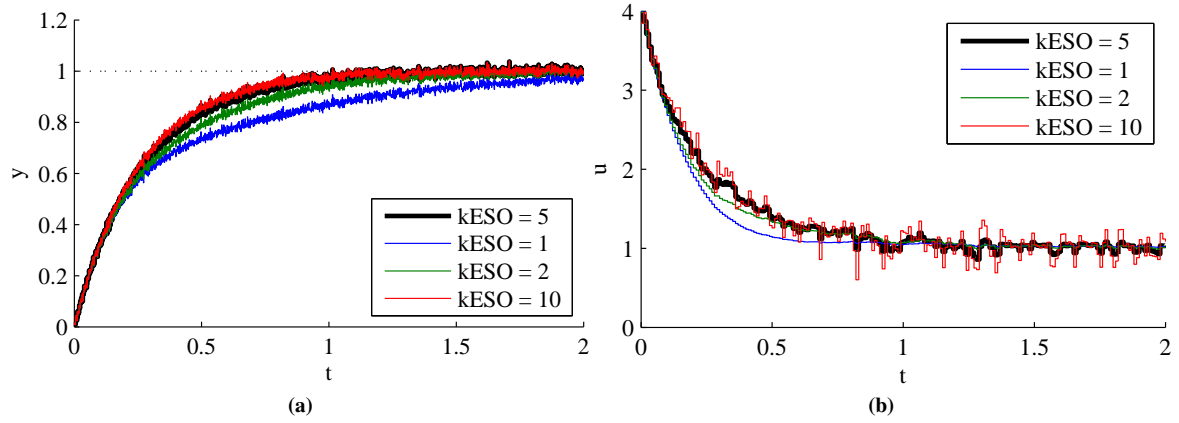

Figure 20. Experiment 4.2.3: influence of observer pole placement via k_{ESO} on discrete-time ADRC. Sample time: $T_{\text{sample}} = 0.01$. Measurement noise variance: $\sigma^2_{\text{noise}} = 0.0001$. (a) Variation of k_{ESO}, closed loop step response; (b) Controller output u for (a).

5   Optimized Discrete-Time Implementation

In a practical discrete-time implementation with tight timing constraints, the lag between sensor input and controller output should be as small as possible, which means that one should try to reduce the computational effort for a controller. Due to the observer-based approach, ADRC has a bigger computational footprint than a classical PID-type controller. In this section, we will try to reduce the number of computations, on the one hand, and present an implementation focused on low input-output delay, on the other hand.

5.1   State Variable Transformation

We start with the discrete-time observer from Equation (23) in the following abbreviated form:

$$\hat{\boldsymbol{x}}(k) = \boldsymbol{A}_{\text{ESO}} \cdot \hat{\boldsymbol{x}}(k-1) + \boldsymbol{B}_{\text{ESO}} \cdot u(k-1) + \boldsymbol{L}_{\text{ESO}} \cdot y(k)$$

and with a control law for the first-order or second-order case:

$$u(k) = \frac{K_{\text{P}} \cdot (r(k) - \hat{x}_1(k)) - \hat{x}_2(k)}{b_0} \quad \text{or} \quad u(k) = \frac{(K_{\text{P}} \cdot (r(k) - \hat{x}_1(k)) - K_{\text{D}} \cdot \hat{x}_2(k)) - \hat{x}_3(k)}{b_0}$$

One can simplify the controller structure by scaling the outputs of the observer, such that the multiplications by b_0, K_{P} and K_{D} (the latter only for the second-order case) can be omitted. The desired scaling of the new state variables, $\widetilde{x}_i$, is achieved via:
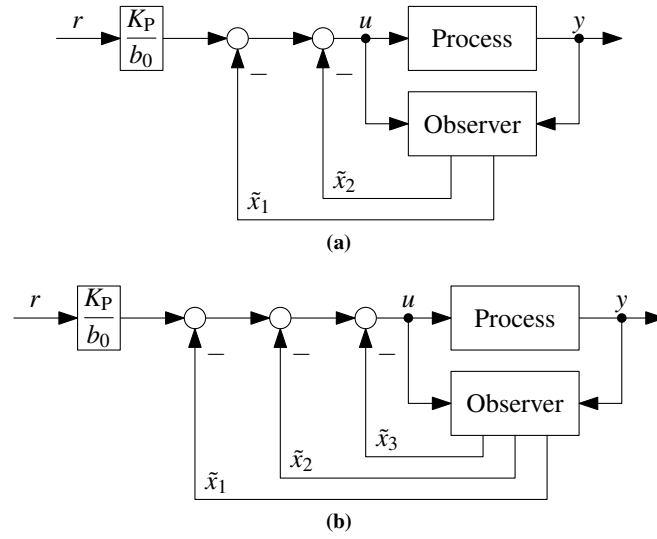
Figure 21. Control loop with modified ADRC structure and transformed state variables, $\tilde{x}$. (**a**) Modified first-order structure; (**b**) Modified second-order structure.

(a) $\tilde{x}_1 = \dfrac{K_P}{b_0} \cdot \hat{x}_1, \quad \tilde{x}_2 = \dfrac{1}{b_0} \cdot \hat{x}_2 \quad$ (for the first-order case)

(b) $\tilde{x}_1 = \dfrac{K_P}{b_0} \cdot \hat{x}_1, \quad \tilde{x}_2 = \dfrac{K_D}{b_0} \cdot \hat{x}_2, \quad \tilde{x}_3 = \dfrac{1}{b_0} \cdot \hat{x}_3 \quad$ (for the second-order case)

The ADRC structure can then be modified as given in Figure 21. Generally speaking, we perform a coordinate transformation from the old estimated variables $\hat{\boldsymbol{x}}$ to $\tilde{\boldsymbol{x}}$ by means of a transformation matrix:

$$\boldsymbol{T}^{-1} = \frac{1}{b_0} \cdot \begin{pmatrix} K_P & 0 \\ 0 & 1 \end{pmatrix} \quad \text{(1}^{\text{st}}\text{ order ADRC)} \quad \text{or} \quad \boldsymbol{T}^{-1} = \frac{1}{b_0} \cdot \begin{pmatrix} K_P & 0 & 0 \\ 0 & K_D & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{(2}^{\text{nd}}\text{ order)} \tag{26}$$

Using this matrix $\boldsymbol{T}$, the state space equations for the extended state observer must be transformed in order to obtain an ESO working with our new state variables, $\tilde{\boldsymbol{x}} = \boldsymbol{T}^{-1} \cdot \hat{\boldsymbol{x}}$:

$$\widetilde{\boldsymbol{A}}_{\text{ESO}} = \boldsymbol{T}^{-1} \cdot \boldsymbol{A}_{\text{ESO}} \cdot \boldsymbol{T}, \quad \widetilde{\boldsymbol{B}}_{\text{ESO}} = \boldsymbol{T}^{-1} \cdot \boldsymbol{B}_{\text{ESO}}, \quad \widetilde{\boldsymbol{L}}_{\text{ESO}} = \boldsymbol{T}^{-1} \cdot \boldsymbol{L}_{\text{ESO}} \tag{27}$$

In this manner, multiplications of the state variables by b_0, K_P and K_D can be avoided. Provided the reference, $r(k)$, does not change at each point in time, one can precompute $\frac{K_P}{b_0} \cdot r$, as well. The control laws are being simplified to the following form:

$$u(k) = \frac{K_P}{b_0} \cdot r(k) - \hat{x}_1(k) - \hat{x}_2(k) \quad \text{or} \quad u(k) = \frac{K_P}{b_0} \cdot r(k) - \tilde{x}_1(k) - \tilde{x}_2(k) - \tilde{x}_3(k)$$

5.2   Minimizing Latency by Precomputation

In an application with fixed sampling frequency, the controller performance can be improved by minimizing the latency between acquisition of input signals and output of the updated controller action within a sampling period. This input-output lag is not necessarily dependent on the overall number of computations of a control law, but rather on the computations necessary to deliver the controller output, with the computations left being performed after that during the remaining time of the sampling period.

We will optimize the discrete-time implementation of ADRC in this regard and, subsequently, derive further equations in detail for the second-order case. Following the ideas of the previous section, the estimated state variables must be updated according to Equation (28) and used to compute the controller output in Equation (29), at each point in time k:

$$\tilde{\boldsymbol{x}}(k) = \widetilde{\boldsymbol{A}}_{\text{ESO}} \cdot \tilde{\boldsymbol{x}}(k-1) + \widetilde{\boldsymbol{B}}_{\text{ESO}} \cdot u(k-1) + \widetilde{\boldsymbol{L}}_{\text{ESO}} \cdot y(k) \tag{28}$$

$$\text{with} \quad \tilde{\boldsymbol{x}} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix}, \quad \widetilde{\boldsymbol{A}}_{\text{ESO}} = \begin{pmatrix} \widetilde{a}_{11} & \widetilde{a}_{12} & \widetilde{a}_{13} \\ \widetilde{a}_{21} & \widetilde{a}_{22} & \widetilde{a}_{33} \\ \widetilde{a}_{31} & \widetilde{a}_{32} & \widetilde{a}_{33} \end{pmatrix}, \quad \widetilde{\boldsymbol{B}}_{\text{ESO}} = \begin{pmatrix} \widetilde{b}_1 \\ \widetilde{b}_2 \\ \widetilde{b}_3 \end{pmatrix}, \quad \widetilde{\boldsymbol{L}}_{\text{ESO}} = \begin{pmatrix} \tilde{l}_1 \\ \tilde{l}_2 \\ \tilde{l}_3 \end{pmatrix}$$

$$u(k) = \frac{K_{\text{P}}}{b_0} \cdot r(k) - \tilde{x}_1(k) - \tilde{x}_2(k) - \tilde{x}_3(k) = \frac{K_{\text{P}}}{b_0} \cdot r(k) - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \tilde{\boldsymbol{x}}(k) \tag{29}$$

When inserting Equation (28) into Equation (29), one can see that $u(k)$ depends both on variables from time k, $y(k)$ and $r(k)$, and from time $k-1$, $u(k-1)$ and $\tilde{\boldsymbol{x}}(k-1)$. An idea for providing $u(k)$ with lower latency would therefore be to precompute all terms stemming from time $k-1$ already at $k-1$ and only update this precomputed value, $u(k|k-1)$, at time k to obtain and output $u(k) = u(k|k)$, as soon as the necessary measurements become available. Further optimization is possible if the reference value, r, does not change rapidly, remains constant or is known in advance. We will denote this by $r(k+1|k)$, *i.e.*, $r(k+1)$ is already known at time k, for any of the reasons mentioned. Then, only the measured process output, $y(k)$, has to be included in the update step to obtain $u(k) = u(k|k)$:

$$u(k) = u(k|k-1) - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \widetilde{\boldsymbol{L}}_{\text{ESO}} \cdot y(k) = u(k|k-1) - \left(\tilde{l}_1 + \tilde{l}_2 + \tilde{l}_3 \right) \cdot y(k) \tag{30}$$

In Equation (30), the sum $(\tilde{l}_1 + \tilde{l}_2 + \tilde{l}_3)$ can be precomputed as well, such that only one multiplication and one addition is necessary to calculate the controller output, $u(k)$. After that, the remaining time of the sampling period can be used to update the observer states via Equation (31) and precompute the output via Equation (32) for the next point in time, $k+1$:

$$\tilde{\boldsymbol{x}}(k+1|k) = \widetilde{\boldsymbol{A}}_{\text{ESO}} \cdot \underbrace{\left(\tilde{\boldsymbol{x}}(k|k-1) + \widetilde{\boldsymbol{L}}_{\text{ESO}} \cdot y(k) \right)}_{\tilde{\boldsymbol{x}}(k)} + \widetilde{\boldsymbol{B}}_{\text{ESO}} \cdot u(k) \tag{31}$$

$$u(k+1|k) = \frac{K_{\text{P}}}{b_0} \cdot r(k+1|k) - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \tilde{\boldsymbol{x}}(k+1|k) \tag{32}$$

Note that the actual observer states, $\tilde{\boldsymbol{x}}(k)$, are not explicitly present and updated in the equations anymore, since only the precomputed values, $\tilde{\boldsymbol{x}}(k+1|k)$, are needed. To summarize, a latency-optimized discrete-time implementation has to perform the following four steps at each point in time k, with the new controller output, $u(k)$, being available already after the second step:

1. Acquire the current measurement of the process output, $y(k)$.

2. Calculate and output $u(k)$ by Equation (30).

3. Update the internal observer states by Equation (31).

4. Precompute the controller output for $k+1$ using Equation (32).

6   Conclusions

By means of simulative experiments using generic first-order and second-order plants, it could be demonstrated that ADRC can be a powerful control tool. In this article, the linear case was examined. It can adapt even to heavily varying process parameters, and—in contrast to "classical" adaptive controllers—do so without having to maintain an explicit model of the process. Since only little knowledge about a process has to be provided along with the desired closed loop dynamics, the parameterization is easy for both continuous time and discrete time cases and, therefore, appealing to practitioners.

For control problems with high dynamic requirements, an optimized formulation of the discrete time linear ADRC equations can be found, which enables the controller output to be computed with only one addition and one multiplication after the sensor input becomes available.

If ADRC can show its full potential in a specific application does, however, depend on the relation of process dynamics, observer dynamics, sampling time and measurement noise. In order to provide the adaptability, the observer has to be fast enough compared to the process and closed loop dynamics, on the one hand. On the other hand, the placement of the observer poles will be limited by sampling frequency and the increasing effects of noise on the control action. As long as a good compromise can be found in this regard, ADRC has to be considered as a strong and welcome alternative to solving practical control problems.

References

[1] Mituhiko Araki and Hidefumi Taguchi. Two-degree-of-freedom PID controllers. *International Journal of Control, Automation, and Systems*, 1(4):401–411, December 2003.

[2] Enrico Canuto. Embedded Model Control: Outline of the theory. *ISA Transactions*, 46(3):363–377, 2007. `doi:10.1016/j.isatra.2007.01.006`.

[3] Xing Chen, Donghai Li, Zhiqiang Gao, and Chuanfeng Wang. Tuning method for second-order active disturbance rejection control. In *Proceedings of the 30th Chinese Control Conference*, pages 6322–6327, 2011.

[4] Bruce A. Francis and Walter Murray Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976. `doi:10.1016/0005-1098(76)90006-6`.

[5] Gene F. Franklin, Michael L. Workman, and Dave Powell. *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing, Boston, MA, USA, 3rd edition, 1997.

[6] Zhiqiang Gao. Scaling and bandwidth-parameterization based controller tuning. In *Proceedings of the 2003 American Control Conference*, pages 4989–4996, 2003. `doi:10.1109/ACC.2003.1242516`.

[7] Zhiqiang Gao. Active disturbance rejection control: A paradigm shift in feedback control system design. In *Proceedings of the 2006 American Control Conference*, pages 2399–2405, 2006. `doi:10.1109/ACC.2006.1656579`.

[8] Zhiqiang Gao, Yi Huang, and Jingqing Han. An alternative paradigm for control system design. In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001. `doi:10.1109/CDC.2001.980926`.

[9] Jingqing Han. From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3):900–906, 2009. `doi:10.1109/TIE.2008.2011621`.

[10] Robert Miklosovic, Aaron Radke, and Zhiqiang Gao. Discrete implementation and generalization of the extended state observer. In *Proceedings of the 2006 American Control Conference*, pages 2209–2214, 2006. `doi:10.1109/ACC.2006.1656547`.

[11] Eric Ostertag. *Mono- and Multivariable Control and Estimation*. Springer, 2011.

[12] S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003. `doi:10.1016/S0967-0661(02)00186-7`.

[13] Aaron Radke and Zhiqiang Gao. A survey of state and disturbance observers for practitioners. In *Proceedings of the 2006 American Control Conference*, pages 5183–5188, 2006. `doi:10.1109/ACC.2006.1657545`.

[14] Y. X. Su, C. H. Zheng, and B. Y. Duan. Automatic disturbances rejection controller for precise motion control of permanent-magnet synchronous motors. *IEEE Transactions on Industrial Electronics*, 52(3):814–823, 2005. `doi:10.1109/TIE.2005.847583`.

[15] Bosheng Sun and Zhiqiang Gao. A DSP-based active disturbance rejection control design for a 1-kW H-bridge DC-DC power converter. *IEEE Transactions on Industrial Electronics*, 52(5):1271–1277, 2005. `doi:10.1109/TIE.2005.855679`.

[16] John Vincent, Dan Morris, Nathan Usher, Zhiqiang Gao, Shen Zhao, Achille Nicoletti, and Qinling Zheng. On active disturbance rejection based control design for superconducting RF cavities. *Nuclear Instruments and Methods in Physics Research A*, 643(1):11–16, 2011. `doi:10.1016/j.nima.2011.04.033`.

[17] Qing Zheng and Zhiqiang Gao. On practical applications of active disturbance rejection control. In *Proceedings of the 29th Chinese Control Conference*, pages 6095–6100, 2010.