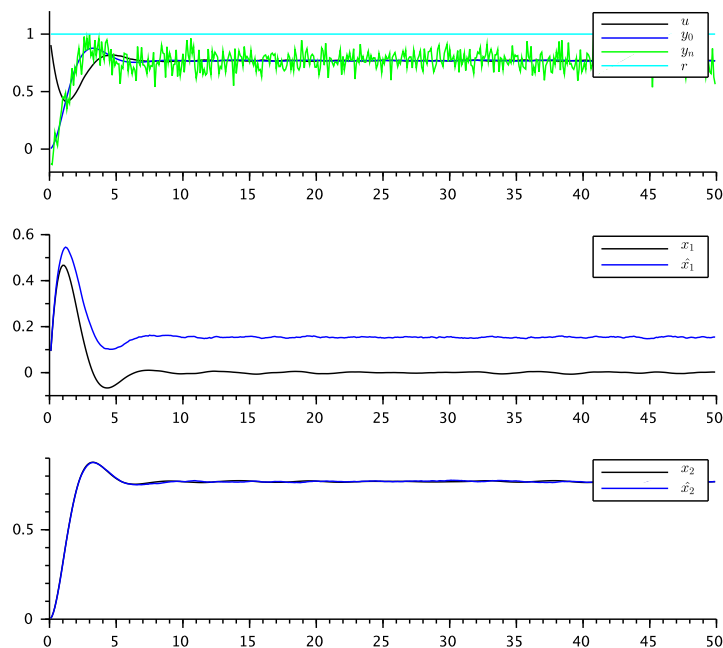


## Praktikum “Zustandsregelung”

**Aufgabe 1** (Zustandsregelung mit Luenberger-Beobachter in Scicos). Ergänzen Sie das System-Modell aus dem zweiten Praktikum um einen Zustandsregler basierend auf den geschätzten Zuständen. Beachten Sie folgende Punkte:

- Basis der Schätzung ist das Ausgangssignal des Beobachters
- Zeichnen Sie folgenden Größen auf:
  - Eingangssignal in das System (die Regelstrecke)  $u$
  - Wahre Zustände  $x_1, y_0 = x_2$
  - Geschätzte Zustände  $\hat{x}_1, \hat{x}_2$
  - Gemessenes (verraushtes) Ausgangssignal  $y_n$
  - Sollwertvorgabe  $r$
- Stellen Sie in je einem Plot gegenüber:
  - $u, y_0, y_n, r$
  - $x_1, \hat{x}_1$
  - $x_2, \hat{x}_2$
- Wählen Sie  $K = (1 \ 0,1)^T$



**Lösung 1** (Skript zur Lösung der Aufgabe 1).

```

clear
delete(gcf())

// load the blocks library and the simulation engine
loadXcosLibs(); loadScicos();
// load diagram
importXcosDiagram("P3.zcos")

N = 1000; // time steps
tmax = 50; // final simulation time
Sigma = 0.1; // Noise amplitude

time = linspace(0,tmax,N)'; // time axis
// Define input data for fromworkspace block
myInput.time = time;
myInput.values = ones(time)+ 0*sin(0.2*time)+0*rand(time);
myNoise.time = time;
myNoise.values = grand(N,1, "nor", 0, Sigma);
// System parameters
a1 = .2;
a0 = 1;
b0 = 1;

A = [0, -a0; 1, -a1];
B = [b0; 0];
C = [0 1];
K = 1*[1, .1];

L = [.2; .1];

// Execute scicos diagram
scicos_simulate(scs_m)
// Plot data
subplot(3,1,1)
plot2d(myOutput.time, myOutput.values(:,[1,3,6, 7]))
legend(' $u$ ', '$y_0$', '$y_n$', '$r$ ')
subplot(3,1,2)
plot2d(myOutput.time, myOutput.values(:,[2,4]))
legend(' $x_1$ ', '$\hat{x}_1$ ')
subplot(3,1,3)
plot2d(myOutput.time, myOutput.values(:,[3,5]))
legend(' $x_2$ ', '$\hat{x}_2$ ')

```

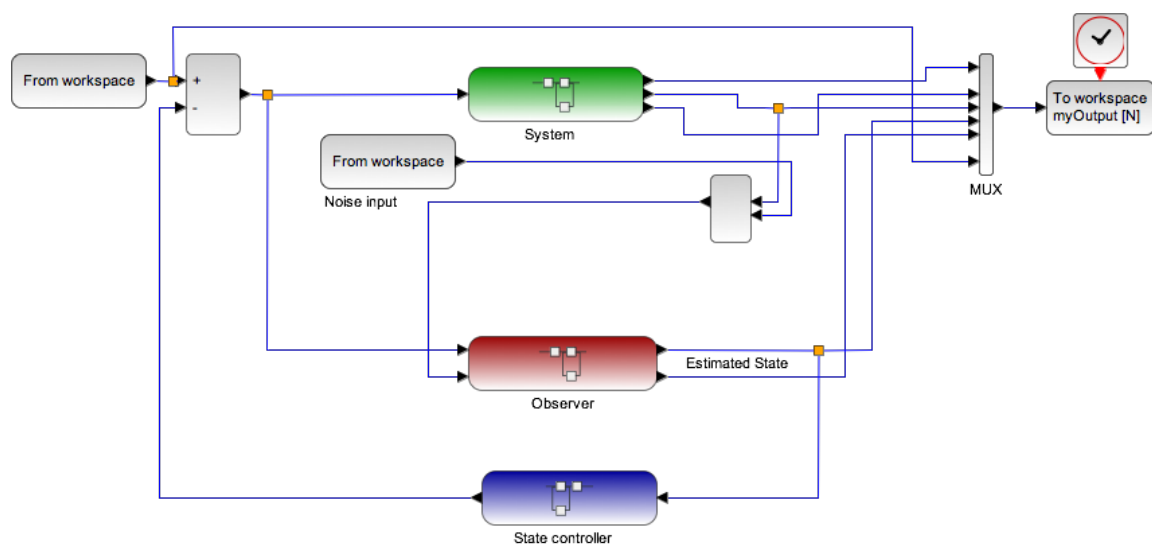


Abbildung 1: XCos Blockdiagramm