A Novel Approach for Design and Analysis of Nonlinear Sampled-Data Control Systems

Dina Shona Laila
d.laila@imperial.ac.uk

Control & Power Group
Electrical & Electronic Engineering Dept.
Imperial College London
What Is A Sampled-Data Control System?

Sampled-data control systems = Computer controlled systems.

General structure:

![Diagram of sampled-data control system]

- A-D: Analog-to-Digital Converter
- Controller
- D-A: Digital-to-Analog Converter
- Plant/Process
- y(t): Output signal
- u(kT): Controller output (sampled)
- u(t): Plant/Process input
What Is A Sampled-Data Control System?

They are very common around us!

Robotic (tele)-surgery.

Flight control.

Space exploration.

Control room.
Motivations

1. Why nonlinear sampled-data control?
   (a) Prevalence of computer-controlled systems.
   (b) Most plants are naturally nonlinear and often the nonlinearities cannot be neglected.
   (c) Some controllers are naturally discrete-time.
   (d) Some design approaches can only be carried out in discrete-time.

2. Why the topic is challenging?
   (a) Sampling often destroys the system’s structure.
   (b) Important properties may not be preserved under sampling.
   (c) Finding a good model for design is not trivial.
   (d) For nonlinear systems the exact discrete-time model is not available.
Possible Design Approaches

- Most plants/processes are continuous-time. Hence, discrete-time controller design is not straightforward.

- Three approaches to design a discrete-time controller:
  - Emulation design:
    - Continuous–time design → Controller discretization → Digital implementation
  - Direct (exact/approximate) discrete-time design:
    - Plant discretization → Discrete–time controller design → Digital implementation
  - Sampled-data design:
    - Continuous–time plant incl. s/h → Discrete–time controller design → Digital implementation
Main Issues

Why direct discrete-time design?

- Emulation design may not be satisfactory in many situations.
- Sampled-data design is too hard when dealing with nonlinear systems.

⇒ Discrete-time design appears to be a possible compromise.

Why use approximate model?

- Exact discrete-time model for nonlinear systems is not available.
- Digital implementation would always involve approximation.

⇒ Approximate models have to be used.
Consider a triple integrator

\[
\dot{x}_1 = x_2; \quad \dot{x}_2 = x_3; \quad \dot{x}_3 = u .
\]

Using Euler model, we obtain a minimum time dead beat controller

\[
u = u_T(x) = \left(-\frac{x_1}{T^3} - \frac{3x_2}{T^2} - \frac{3x_3}{T}\right).
\]

Ad hoc design does not guarantee that the stability of the approximate model implies stability of the sampled-data system.
The framework provides sufficient conditions to guarantee that a stabilizing controller designed using an approximate model will also stabilize (in a weaker sense) the exact model.

Consider a general nonlinear plant:

\[ \dot{x}(t) = f(x(t), u(t), w(t)). \]

Family of exact discrete-time models:

\[ x(k + 1) = F^e_T(x(k), u(k), w[k]). \]

Family of approximate discrete-time models:

\[ x(k + 1) = F^a_T(x(k), u(k), w(k)). \]

Family of dynamic feedback controllers:

\[ z(k + 1) = G_T(x(k), z(k)); \quad u(k) = u_T(x(k), z(k)). \]
We consider robust stability property, e.g. input to state stability, in a semiglobal practical sense (SP-ISS).

Semiglobal practical (SP) stability:

\[ S \]
Suppose the following hold:

**SP-ISS Lyapunov character.**

for closed-loop approx.: \[ V_T(F_a T) \leq V_T(x) \leq V_T(F_a T) - \frac{\alpha_3(|\tilde{x}|)}{T} + \tilde{\gamma}(\|w\|_{\infty}) + \delta_1 \]

Local Lipschitzity of \( V_T \) in \( x \):
\[ |V_T(x_1, z) - V_T(x_2, z)| \leq L |x_1 - x_2| \]

One step consistency:
\[ |F^e_T - F^\alpha_T| \leq T \rho(T) \]

Uniform local boundedness of \( u_T \):
\[ |u_T(\tilde{x})| \leq \Delta_u \]

Then the following holds:

**SP-ISS of closed-loop exact:** \[ |\tilde{x}(k)| \leq \beta(|\tilde{x}(0)|, kT) + \gamma(\|w\|_{\infty}) + \delta \quad \forall k \in \mathbb{N} \]

**All conditions are checkable!**
Design Tools Within The Framework

(a) Euler based discrete-time backstepping design [Nešić & Teel].

- For systems in feedback form.
  \[
  \begin{align*}
  \dot{x} &= f(x) + g(x)\xi \\
  \dot{\xi} &= u .
  \end{align*}
  \]

(b) Perturbed homogeneous time-varying controller.

- For systems in power form.
  \[
  \begin{align*}
  \dot{x}_1 &= u_1 + g_1(x, d) \\
  \dot{x}_2 &= u_2 + g_2(x, d) \\
  \dot{x}_3 &= x_1u_2 + g_3(x, d) \\
  \dot{x}_4 &= \frac{1}{2}x_1^2u_2 + g_4(x, d) \\
  \dot{x}_n &= \frac{1}{(n-2)!}x_1^{n-2}u_2 + g_n(x, d).
  \end{align*}
  \]
(c) Discrete-time IDA-PBC design.

- For Hamiltonian systems.
  \[
  \dot{q} = \nabla_p H(q, p) \quad \dot{p} = -\nabla_q H(q, p)
  \]
  with Hamiltonian function
  \[
  H(q, p) = p^\top M^{-1}(q)p + P(q) .
  \]

(d) Lyapunov-based redesign using Fliess series expansion [Nešić & Grüne].

- For control affine systems.
  \[
  \dot{x} = f(x) + g(x)u .
  \]
The framework requires knowledge of a strict Lyapunov function for the closed-loop approximate.

Obstacles:

- It is in general not easy to find a strict Lyapunov function for nonlinear systems.
- For semiglobal practical stability, La Salle Invariance Principle is not applicable.

Techniques to construct a strict Lyapunov function for classes of nonlinear systems are important.
(a) Changing supply rates [SP-qISS + SP-IOSS ⇒ SP-ISS]

For systems with partial state output

\[ x_{k+1} = F_T(x_k, u_k) \]
\[ y_k = h_T(x_k) . \]

If \( V_{1T} \) is a SP-qISS Lyapunov function (La Salle)
\( V_{2T} \) is an SP-IOSS Lyapunov function (detectability),
then

\[ V_T = V_{1T} + \rho(V_{2T}) , \quad \rho \in \mathcal{K}_{\infty} \]

is an SP-ISS Lyapunov function for the system.
Design Framework (con’t)

Suppose the following hold:

SP-ISS Lyapunov charact. for closed-loop approx.: 
\[
\alpha_1(|\tilde{x}|) \leq V_T(\tilde{x}) \leq \alpha_2(|\tilde{x}|)
\]
\[
\frac{V_T(F_T^0) - V_T(\tilde{x})}{T} \leq -\alpha_3(|\tilde{x}|) + \tilde{\gamma}(\|w\|_{\infty}) + \delta_1
\]

Local Lipschitzity of \( V_T \) in \( x \):
\[
|V_T(x_1, z) - V_T(x_2, z)| \leq L |x_1 - x_2|.
\]

One step consistency:
\[
|F_T^e - F_T^o| \leq T \rho(T).
\]

Uniform local boundedness of \( u_T \):
\[
|u_T(\tilde{x})| \leq \Delta_u.
\]

Then the following holds:

SP-ISS of closed-loop exact:
\[
|\tilde{x}(k)| \leq \beta(|\tilde{x}(0)|, kT) + \gamma(\|w\|_{\infty}) + \delta, \ \forall k \in \mathbb{N}
\]

All conditions are checkable!
(b) Small Gain Theorem

Consider parameterized interconnected systems:

\[ \Sigma_1 : x_1(k + 1) = F_1T(x_1(k), x_2(k), u(k)) , \]
\[ \Sigma_2 : x_2(k + 1) = F_2T(x_1(k), x_2(k), u(k)) . \]

Suppose:

A1. \( \Sigma_1 \) is SP-ISS with inputs \( x_2 \) and \( u \), and SP-ISS Lyapunov function \( V_{1T} \).
A2. \( \Sigma_2 \) is SP-ISS with inputs \( x_1 \) and \( u \), and SP-ISS Lyapunov function \( V_{2T} \).
A3. Small gain condition is satisfied.

Then \( (\Sigma_1, \Sigma_2) \) is SP-ISS with respect to \( u \), and

\[ V_T(x_1, x_2) := \max\{V_{1T}(x_1), \rho(V_{2T}(x_2))\} , \quad \rho \in \mathcal{K}_{\infty} \]

is a SP-ISS Lyapunov function for the interconnected system.
Most of design tools propose state feedback control design. This assumes the availability of all states through measurements. In reality, it is often the case that only a part of the states are available. Several reasons:

- some of the states are not measurable.
- some sensors are very expensive.
- the use of many sensors takes space, energy and weight.
- environmental conditions that does not allow a sensor to do a good sensing.

Observer is then used to estimate the unmeasured states, and the control law is implemented using output feedback.
(a) Euler based observer with inverse mapping.

For nonlinear affine systems.

\[
\begin{align*}
\dot{\eta} &= f_1(\eta)\xi \\
\dot{\xi} &= f_2(\eta, \xi) + g(\eta)u \\
y &= \eta,
\end{align*}
\]

Euler approximate model:

\[
\eta_{k+1}^a = \eta_k^a + T f_1(\eta_k^a)\xi_k^a.
\]

Inverse mapping estimation:

\[
\begin{align*}
\dot{\xi}_k^a &= \left( f_1^T(\eta_k^e) f_1(\eta_k^e) \right)^{-1} f_1^T(\eta_k^e) \eta_{k+1}^e - \eta_k^e \\
 &= f_1^+(\eta_k^e) \frac{\eta_{k+1}^e - \eta_k^e}{T}.
\end{align*}
\]
(b) Orthogonal observer.

For spacecraft systems.

\[ \dot{\omega} = J^{-1}(-\omega \times J\omega) + u \]
\[ \dot{A} = S(\omega)A \]
\[ y_1 = \omega; \ y_2 = \tilde{b} . \]

Orthogonality preserving model:

\[ A_{k+1} = A_k + TS(\omega_k) \frac{1}{2} (A_k + A_{k+1}) \]
\[ y_{1k} = \omega_k; \ y_{2k} = \tilde{b}_k . \]

Orthogonal observer:

\[ \hat{A}_{k+1} = \hat{A}_k + T \left[ \frac{S(\omega_k)}{2} (\hat{A}_k + \hat{A}_{k+1}) + (b_k - \hat{b}_k) L^T R_k^\top \right] . \]
Application Example 1

Discrete-time IDA-PBC design.

Inverted pendulum:

Other applications: robotic manipulator, aircraft, underwater vehicle, electrical network, magnetic levitation and other mechanical and electromechanical systems.
Application Example 1 (cont’d)

Lyapunov function construction:

\[ \text{SP-qISS} + \rho \circ \text{SP-IOSS} \]

SP-ISS
Application Example 2

Perturbed homogeneous time-varying controller.

Unicycle mobile robot:

Other applications: car, truck, train and other vehicles with trailers.
Application Example 3

Spacecraft attitude control with output feedback.

Space explorer:

Other applications: space shuttle, aircraft, satellite and other spacecrafts.
1. Summary

- Sampled-data control systems follow the recent development of computer technology and the need of automation process.
- (Approximate) direct discrete-time design promises a balance between the achievable performance and design complexity.
- (Approximate) direct discrete-time design increases the degree of freedom in design, by playing with various discrete-time models.

2. Open Problems

- Develop more constructive results for various classes of systems.
- Explore non-Lyapunov based approaches.
- Develop results that guarantee global stability or large domain of attraction for not necessarily fast sampling.
- Construct design techniques that allow to compute quantitatively the improvement of performance.
- Develop more results for observer design and output feedback control design.
- Applying the results to real life applications, hardware implementation, and many others....
Main Collaborators & Contributors

- Alessandro Astolfi (Imperial College London, UK).
- Dragan Nešić (University of Melbourne, Australia).
- Andrew R. Teel (University of California, Santa Barbara, USA).
- Lars Grüne (University of Bayreuth, Germany).
- Marco Lovera (Politecnico di Milano, Italy).