The Aim of This Research

- To investigate the development of numerical methods for systems and control which have a guarantee on accuracy.
- An end-product — an “infallible” algorithm: the user would specify a priori a tolerance as small as desired, and the computer would provide an answer which was guaranteed to be accurate to the specified tolerance.
- An established subject within Computer Science and a few application areas in science and engineering. Quite a new direction in the systems and control area.

A characteristic feature — the application of computer algebra tools and the avoidance of floating-point arithmetic.

Problems Investigated

- Computation of certain quantities widely used in the modern analysis methods for control system performance, e.g. the $H_2$ and $H_{\infty}$ norms of a stable transfer function and the induced norm of a linear system.
- Some common algorithms for controller synthesis problems, e.g., controller computation for $H_2$ and $H_{\infty}$ optimisation.

An Example: $L_\infty$ Norm Computation

If the computation of a scalar quantity, e.g., the $L_\infty$ norm, is sought, a “validated numerical method” will produce an interval of a user-specified width within which the numerical answer is guaranteed to lie.

The $L_\infty$ norm of $G(s)$, a matrix of rational functions of the complex variable $s$, is defined as $\|G(s)\|_\infty := \sup_{s \in \mathbb{C}} \tau(G(j\omega))$ where $\tau(\cdot)$ is the largest singular value. A current implementation of the $L_\infty$ norm computation in Matlab (a software package commonly used in systems and control area) uses floating-point arithmetic and is prone to numerical error. For instance, it fails to compute the $L_\infty$ norm of $G(s) = \frac{s^2 + 10^{-7}s + 1}{s^2 + 10^{-3}s + 1}$ whose $L_\infty$ norm is $\|G(s)\|_\infty = 10$. 

> out = linfnorm(nd2sys([1 10^-7 1], [1 10^-8 1]))

LINFNORM iteration DID NOT converge to a lower bound for the norm is 10
out = 10.000000001492336
Inf = 1.000000000000000

A validated numerical algorithm implemented in Maple makes use of the result below.

**Theorem 1 ([1])**

Let $G(s) \in L_\infty$ be rational and write its $L_\infty$ norm $\|G(s)\|_\infty$ as $\gamma_\infty$. Let $\Phi_1(s) = s^2I - G^T(-s)G(s)$ and denote $g_i(s) = \det \Phi_i(s)$. Moreover, write $g_i(x) = \frac{\gamma_i(x)}{\gamma_i}$. Let $h_i(x)$ be the square-free part of $n_i(x)$ considered as a polynomial in $x$ and $\gamma$. Then, $\gamma > \gamma_\infty$ if and only if $\gamma > \tau_i(G(j\infty))$ and $h_i(x)$ has no roots in $-\infty < x \leq 0$. Further, if $\gamma_\infty$ is achieved in $0 < \omega < \infty$, then $h_i(x)$ has a multiple root in $-\infty < x < 0$.

- The norm is equal to $\tau(G(0)), \tau(G(j\infty))$ or a real root of the discriminant of $h_i(x)$.
- The $L_\infty$ norm computation of $G(s)$ reduced to the univariate polynomial real root computation problem and the Sturm test.
- That is, intervals with arbitrary widths which contain these candidates can be found by means of standard real root localisation methods, for instance, Descartes’ rule of signs.
- The Sturm test can determine which interval contains the true $L_\infty$ norm by examining whether $\gamma > \tau_i(G(j\infty))$ and also the existence of roots of $h_i(x)$ in $-\infty < x \leq 0$.

For the above example, the actual $L_\infty$ norm is found from the discriminant of $h_i(x)$, $(\gamma + 10)(\gamma - 10)(133333333333333333333333333333333333)$.

We emphasise that the original real number data, and each step of the test, makes use of rational number arithmetic only. Rounding errors are avoided and the method counts as a validated numerical method. This contrasts with all current implementations of the $L_\infty$ norm which can suffer from numerical error.

Controller Synthesis Problems

For instance, the $H_2$ optimal controller synthesis problem is formulated as: Given a plant $P$, find a controller $K$ that stabilises the closed-loop system and minimise the $H_2$ norm of the transfer function from $(d_1, d_2)^T$ to $(y_1, y_2)^T$.

The development of validated numerical methods for the computation of controllers using some standard synthesis procedures presents a number of interesting challenges.

- May not always be desirable to specify “guaranteed accuracy” in terms of the constants of the controller.
- A possibility: Find a controller whose distance from the actual controller is within a user-specified value in terms of the standard metrics on dynamical systems, such as the gap or $\nu$-gap metric.
- Questions relating to continuity of solution to be answered along with algorithm development.
- A multiple stage algorithm may require use of interval methods.

Progress in this direction has been reported in [1].

Further Research

- Tackle problems which do not allow satisfactory algorithms to be implemented using ordinary floating-point arithmetic.
- Lack of reliable computational tools has been preventing some theoretical developments from being used in practice. Investigation of these problems is crucial and could have a significant impact.

References