EECS 562/AE 551 - Nonlinear Systems and Control
HW #2

Due on January 19th, 2023
by 11:59pm, on Canvas

Note: You should be diligently reading Chapter 3 of Khalil.

Note: Hints for problems are given on a separate page.

1. (20 points) For any $x \in \mathbb{R}^n$, show that

ROB 501

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\,\|x\|_2$$
$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\,\|x\|_\infty$$
$$\|x\|_\infty \leq \|x\|_1 \leq n\,\|x\|_\infty$$

Recall the inequality, $2ab \leq \left(a^2 + b^2\right)$. In the above, you may wish to try the case $n = 2$ first. In fact, working correctly the case $n = 2$ will earn full credit.

2. (20 points) Prove that if $f : \mathbb{R} \longrightarrow \mathbb{R}$ is continuous at x_0, and if $\lim_{n\to\infty} x_n = x_0$, then $\lim_{n\to\infty} f(x_n) = f(x_0)$. ROB 501

3. (20 points) Starting from the definition of a closed set given in the handout (i.e., S is closed $\Longleftrightarrow \sim S$ is open), prove the following:

   (a) Let $(V, \|\cdot\|)$ be a normed space and let $S \subset V$ be a closed subset of V. Let (x_n) be a sequence of elements of S (i.e., $x_n \in S\ \forall n \geq 1$) and suppose that $\lim_{n\to\infty} x_n = \bar{x}$. Then $\bar{x} \in S$. (one says that if S is closed, it contains its limit points) definition of seq and close set

   (b) (not required to turn this part in) Conversely, if S is non-empty and not closed, then there exists a sequence of elements of S which converges to a point $\bar{x} \notin S$.

   Remark: This proof is more challenging than the previous problem. Recall that we do not do proofs on exams, so if you do not get the proof, it is not that big of a deal. The main reason we do a few proofs in HW is that writing a proof causes you to learn all of the definitions we have covered.

4. (20 points) For the following systems, state and prove if the solutions would exist for all $t \in [0, \infty)$ for all initial conditions:

   (a) $\dot{x} = -x^2$ nope, locally liptcz but not contraction mapping

   (b) $\dot{x} = -x^3$ yes, contraction mapping

   (c) $\dot{x} = x^2 \cos(x^3)$ yes, contraction mapping

   (d) $\dot{x} = x\,\mathrm{sat}(x^4)$ disc

5. (20 points) The goal of this problem is to actually use the Contraction Mapping Principle to compute the solution of an ODE. We will consider the scalar ODE, $\dot{x} = \sin(x)$. Note that $f(x) = \sin(x)$ is globally Lipschitz continuous with Lipschitz constant $L = 1$;

   (a) Find a $\delta > 0$ such that $P : C[0, \delta] \longrightarrow C[0, \delta]$ is a contraction, where

$$P(\varphi)(t) := x_0 + \int_0^t f(\varphi(\tau))\, d\tau$$

2*pi?

   and the norm is the max norm on $C[0, \delta]$.

(b) Write script in MATLAB (or use any other programming language) that implements the Contraction Mapping Principle to numerically compute a solution to the ODE on the interval $[0, \delta]$ with $x_0 = 1$; this will really test whether you understand what we have done, so be persistent. Plot your computed solution. ok

(c) Solve the ODE using any numerical ODE solver and make a plot of your answer. Compare this answer to what you computed in part (b). ok

Turn in your script and a plot (or plots) comparing your answers in (5c). My hope is that you realize that much of the mathematics we do in EECS 562 can actually be implemented in code.

To think about: How would you use the Contraction Mapping Principle to compute the solution over a given interval of time, such as $[0, 10]$? (You do NOT need to answer this as part of your HW solution.)

Hints

Hints for Problem 1: It is easiest to start with $||x||_\infty \le K||x||_1$ and find K. The next easiest one is $||x||_1 \le K||x||_\infty$ and find K,
Note: When working with the 2-norm, it is often easier to work with its square. For example, proving

$$\|x\|_2 \le K \|x\|_1$$

is equivalent to proving

$$\|x\|_2^2 \le K^2 \|x\|_1^2$$

Note: Because $(a-b)^2 \ge 0$, expanding it out shows that $2ab \le 2|a| \cdot |b| \le a^2 + b^2 \le 2(a^2+b^2)$. This inequality may be useful to you when dealing with the 2-norm.

Hints for Problem 2: Let $y_n = f(x_n)$ and $y_0 = f(x_0)$. Write down what you need to show. Then write down what you know from the continuity of f at x_0 and the convergence of (x_n) to x_0. The proof will then be much easier.

Hints for Problem (3a): Suppose you want to show that $p \Rightarrow q$. Most of you know that $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$. This is called a proof by *contraposition*. The easiest way to work this particular HW problem is to employ a different kind of proof that many of you do not know well, namely a *proof by contradiction*. Specifically, assume that S is a closed set, the sequence (x_n) satisfies that $x_n \in S$ for all $n \ge 1$ AND that $\bar{x} \notin S$. Then prove that it cannot be the case that $\lim_{n\to\infty} x_n = \bar{x}$.

You can also view this as a proof by contraposition if you formulate the problem like this. Let S be a closed set and let (x_n) be a sequence of elements in S. If **(p)** $\lim_{n\to\infty} x_n = \bar{x}$, then **(q)** $\bar{x} \in S$.

Further hint: Note that if $\bar{x} \notin S$, then $\bar{x} \in \sim S$, the set complement of S. Furthermore, observe that $\sim S$ is open because S is closed. Draw a picture before you start your proof; indeed, draw a small open ball around $\bar{x} \in \sim S$ and recall that $x_n \in S$.

Hint for Problem (4a) and (4b): Try finding the solution of the differential equation by integrating it.

Hint for Problem (4c): Find the equilibrium(s) of the system.

Hints for Problem 5:

Conceptual Remark: When solving an ODE, the Contraction Mapping Principle involves iterating on functions. What makes this problem hard for many of you is that you fail to see how MATALB can iterate on a function. The way to think about this is that MATLAB is very good at vectors. Consider a function $g : [0, 2\pi] \to \mathbb{R}$, such as $g(t) = \sin(t)$, which is clearly recognized as a function. Suppose you wanted to `plot` this function in MATLAB. You would execute something like the first half of the code below, which gives the top plot. Now look at the second half of the code, which plots only the DOTS and does not connect them with lines. Now you see only the values of the function $\sin(t)$ at the discrete points specified by $t = [0, 0.05, 0.1, \cdots, 2\pi]$. Hence, a function in MATLAB can be viewed as a vector once you evaluate it at a finite set of points. When you program up the Contraction Mapping Principle in MATLAB, you will be computing the solution of the ODE (a function) at discrete points of time (giving you a vector), just as in the plots below! More detailed hints follow.

```
dt=0.05;
t=[0:dt:2*pi]

g=sin(t);

figure(1)
subplot(2,1,1)
hand=plot(t,g)
xlabel('time (s)')
ylabel('g(t)')
set(hand,'LineWidth',2)
grid on
```

```
subplot(2,1,2)
hand=plot(t,g,'.')
xlabel('time (s)')
ylabel('g(t)')
set(hand,'LineWidth',2)
grid on

print fig_sine_function -dpng
```

Specific Hints:

(i) Discretize the time interval $[0, \delta]$, for example, as $[t_0, t_1, t_2, \ldots, t_N]$, where $t_0 = 0$, $t_N = \delta$, and $(t_k - t_{k-1}) = dt = \delta/N$

(ii) Approximate the integral in the definition of P as a finite sum, using the definition of a Riemann integral:

$$\int_{t_0}^{t_k} f(\varphi(\tau))d\tau \approx \begin{cases} 0 & t_k = t_0 \\ \sum_{j=1}^{k} f\left(\varphi\left(t_j\right)\right) \frac{\delta}{N} & t_k > t_0 \end{cases}.$$

This will lead you to,

$$\begin{aligned} P\left(\varphi\right)\left(t_0\right) &= x_0 \\ P\left(\varphi\right)\left(t_k\right) &= x_0 + \sum_{j=1}^{k} f\left(\varphi\left(t_j\right)\right) \frac{\delta}{N} \\ &= P\left(\varphi\right)\left(t_{k-1}\right) + f\left(\varphi\left(t_k\right)\right) \frac{\delta}{N}, \quad k = 1, \ldots, N \end{aligned}$$

(iii) Initialize the iteration $\varphi_{n+1} = P\left(\varphi_n\right)$ with the constant function $\varphi_0\left(t_k\right) = x_0$, $t_k \in [t_0, t_1, t_2, \ldots, t_N]$.

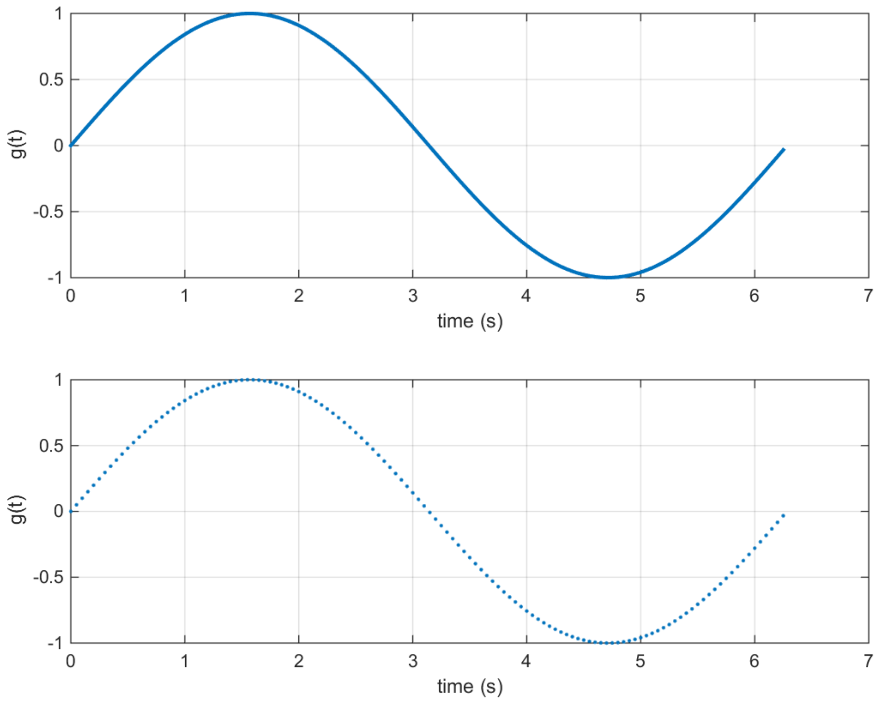

Figure 1: **Functions in MATLAB can be thought of as vectors:** The top figure is your standard plot of $\sin(t)$ for $t \in [0, 2\pi]$, while the second plot is $\sin(t)$ at discrete values of the same interval. The second plot is how to think of functions in MATLAB.

(iv) From hint (ii), note that

$$
\begin{aligned}
\varphi_{n+1}(t_k) &= P(\varphi_n)(t_k) \\
&= x_0 + \sum_{j=1}^{k} f(\varphi_n(t_j)) \frac{\delta}{N} \\
&= P(\varphi_n)(t_{k-1}) + f(\varphi_n(t_k)) \frac{\delta}{N}, \quad k = 1, \ldots, N
\end{aligned}
$$

Now, build your MATLAB code

You are likely to have nested FOR LOOPS: an `outer loop` indexed by n implementing $\varphi_{n+1} = P(\varphi_n)$, that iterates the map P in the Contraction Mapping Principle, and an `inner loop` evaluating your function at each discrete point in the interval $[0, \delta]$, namely $\varphi_{n+1}(t_k) = P(\varphi_n)(t_k) = x_0 + \sum_{j=1}^{k} f(\varphi_n(t_j)) \frac{\delta}{N}$, where the sum is implementing the approximate definition of the Riemann integral at the discrete time points $t_k \in \{t_0, t_1, \ldots, t_N\}$.