Homework 1
ESE 601, Spring 2006

Truong Xuan Nghiem
nghiem@seas.upenn.edu

Problem 1

1. We have:

- $U_0 = V$ which is a subspace of $\mathbb{R}^n$. Thus, U_0 is a vector space
- Assume that U_i is a vector space, for $i \geq 0$. We will show that $U_{i+1} = U_i \cap A^{-1}U_i$ is also a vector space. From the definition of U_{i+1}, it is clear that $U_{i+1} \subseteq U_i$. We also have:
  - For any $x, y \in U_{i+1}$: $x, y \in U_i \cap A^{-1}U_i$, thus $x \in U_i$ and $x \in A^{-1}U_i \Rightarrow Ax \in U_i$. Similarly, $y \in U_i$ and $Ay \in U_i$. Since U_i is a vector space, it follows that $x + y \in U_i$ and $Ax + Ay = A(x + y) \in U_i$. Therefore, $x + y \in U_i \cap A^{-1}U_i = U_{i+1}$.
  - For any number $a \in \mathbb{R}$ and $x \in U_{i+1}$: because $x \in U_i$, $Ax \in U_i$, and U_i is a vector space, we have $ax \in U_i$ and $aAx = A(ax) \in U_i$, which means $ax \in U_i \cap A^{-1}U_i = U_{i+1}$

  Thus, $U_{i+1} \subseteq U_i$ is a vector space.

By induction on i, it follows that U_i, where $i \geq 0$, are vector spaces.

Since, for any $i \geq 0$, $U_{i+1} \subseteq U_i$, and U_{i+1} and U_i are vector spaces, $\dim(U_{i+1}) \leq \dim(U_i)$, i.e. the dimensions of the sequence of spaces U_i are non-increasing. There are only two cases:

- If U_{i+1} is a proper subset of U_i, then $\dim(U_{i+1}) < \dim(U_i)$. This can be proved by contradiction. Assume $\dim(U_{i+1}) = \dim(U_i) = k$, then there exists a basis $\{u_1, u_2, \ldots, u_k\}$ of U_{i+1} such that $U_{i+1} = \text{span}\{u_1, u_2, \ldots, u_k\}$. Since $\dim(U_i) = k$ and $u_1, u_2, \ldots, u_k \in U_i$, $\{u_1, u_2, \ldots, u_k\}$ is also a basis of U_i. Thus $U_i = \text{span}\{u_1, u_2, \ldots, u_k\} = U_{i+1}$, which contradicts the hypothesis that $U_{i+1} \subset U_i$.
- If $U_{i+1} = U_i$, then it must be that $U_{i+2} = U_{i+1}$, $U_{i+3} = U_{i+2}$, and so on. In other words, for all $j \geq i$, $U_{j+1} = U_j = U_i$

Since $\dim(V)$ is finite, it follows that the sequence $\dim(U_i)$, for $i \geq 0$, is decreasing until the first finite index N such that $U_{N+1} = U_N$, after which we have $U_{j+1} = U_j$, $\forall j \geq N$, and the iteration terminates. In other words, the iteration of the algorithm terminates after a finite number of steps. □

2. The fix point of the iteration is the index N, corresponding to the vector space U_N.

Since $U_{N+1} = U_N \cap A^{-1}U_N = U_N$, we have $U_N \subseteq A^{-1}U_N$. Thus, for any $x \in U_N$, $x \in A^{-1}U_N$, which means $Ax \in U_N$. Therefore $AU_N \subseteq U_N$, or equivalently, U_N is invariant under A. □

3. We will prove that U_N is the largest invariant subspace in V by contradiction.

Assume that U_N is not the largest invariant subspace in V, i.e. there exists a subspace $W \subseteq V$ such that

- W is A-invariant: $AW \subseteq W$
- $W \nsubseteq U_N$

The vector space W may be a superset of U_N or not. However, we only need to consider the case when $W \supset U_N$ since for the case $W \not\supset U_N$, we can always define $W' = U_N \cup W$ which is a superset of U_N and is invariant under A in V (because for all $x \in W'$, $x \in U_N$ or $x \in W$, thus Ax must be in U_N or W, which means $Ax \in W'$ or $AW' \subseteq W'$).

We've shown in part 1 that $\dim(U_i)$, where $i \geq 0$, is decreasing until $i = N$. Also $\dim(W) > \dim(U_N)$ since U_N is a proper subset of W. Thus, there are only two cases that may happen

- There exists $M < N$ such that $U_M = W$, i.e. W is in the sequence U_i produced by the iteration. However, as shown in part 1, the iteration must stop at $i = M$ and $U_i = U_M = W$ for all $i > M$, which contradicts the existence of U_N; or
- The vector space W is not in the sequence U_i produced by the iteration. In this case, because $W \subset V$ (W cannot equal V, otherwise $W = U_0$ which is in the sequence) and $\dim(U_i)$ is decreasing, there must exist $M \geq 0$ such that $M < N$ and $U_M \supset W$, but $U_{M+1} \subset W$. For every x in W, we have $x \in U_M$ and $Ax \in W \Rightarrow Ax \in U_M$ which means $x \in A^{-1}U_M$. Thus, $x \in U_M \cap A^{-1}U_M = U_{M+1}$. Therefore, $W \subseteq U_{M+1}$ which contradicts the fact that U_{M+1} is a proper subset of W.

By proof by contradiction, it follows that U_N is the largest A-invariant subspace of V. $\square$

Problem 2 To prove $L_2 = L_1^* L_3$, we will show that $L_2 \subseteq L_1^* L_3$ and $L_2 \supseteq L_1^* L_3$.

Prove $L_2 \subseteq L_1^* L_3$: Given any string w in L_2. Since $L_2 = L_1 L_2 \cup L_3$, w must be in L_3 or in $L_1 L_2$.

If $w \in L_3$ then it is in $L_1^* L_3$.

Otherwise, w is in $L_1 L_2$ and it can be decomposed into two strings, $w_1 \in L_1$ and $w_1^\flat \in L_2$, such that $w = w_1 w_1^\flat$. String w_1 must be non-empty since $\epsilon \notin L_1$, however $w_1^\flat$ may be empty. Because $w_1 \neq \epsilon$, $|w_1^\flat| < |w|$ (where $|\,\bullet\,|$ denotes the length of a string). Perform the following algorithm: for $k = 1, 2, \ldots$ and $w_k^\flat \in L_2$

- If $w_k^\flat = \epsilon$ or $w_k^\flat \in L_3$ then the iteration stops.
- Otherwise, $w_k^\flat$ must be in $L_1 L_2$ and it can be decomposed into two strings, $w_{k+1} \in L_1$ and $w_{k+1}^\flat \in L_2$, such that $w_k^\flat = w_{k+1} w_{k+1}^\flat$. String w_{k+1} is non-empty (because $\epsilon \notin L_1$), thus $|w_{k+1}^\flat| < |w_k^\flat|$. Repeat the iteration for $w_{k+1}^\flat$.

Because $|w_k^\flat|$ is strictly monotonically decreasing, and $|w_k^\flat| < |w|$, and $|w|$ is finite, the algorithm must terminate after a finite number of steps, resulting in $w = w_1 w_2 \ldots w_{m-1} w_m^\flat$ where $w_1, w_2, \ldots, w_{m-1}$ are in L_1 and $w_m^\flat$ is either in L_3 or empty. If $w_m^\flat = \epsilon$ then L_3 must contain ϵ, i.e. $w_m^\flat \in L_3$, because otherwise, $\epsilon = w_m^\flat \in L_2 = L_1 L_2 \cup L_3 \Rightarrow \epsilon \in L_1 L_2 \Rightarrow L_1 \ni \epsilon$ which contradicts the assumption that L_1 does not contain the empty string. Therefore, $w \in L_1^{m-1} L_3 \Rightarrow w \in L_1^* L_3$.

It follows that $w \in L_1^* L_3$ for all $w \in L_2$. In other words, $L_2 \subseteq L_1^* L_3$.

Prove $L_2 \supseteq L_1^* L_3$: we have $L_1^* L_3 = \bigcup_{k \geq 0} L_1^k L_3$, thus we only need to show that $L_1^k L_3 \subseteq L_2$ for all $k \geq 0$. We will prove this by induction on k.

- Basis: When $k = 0$, we have $L_1^k L_3 = L_3 \subseteq L_2$ since $L_2 = L_1 L_2 \cup L_3$.
- Induction: Assume that $L_1^k L_3 \subseteq L_2$, where $k \geq 0$, we need to prove that $L_1^{k+1} L_3 \subseteq L_2$. For any string $w \in L_1^{k+1} L_3$, w can be written as $w = w_1 w_2$ where $w_1 \in L_1$ and $w_2 \in L_1^k L_3$. It follows that $w_2 \in L_2$ and $w \in L_1 L_2$. Because $L_1 L_2 \subseteq L_2$, w must be in L_2. Thus, $L_1^{k+1} L_3 \subseteq L_2$.

Therefore, $L_1^k L_3 \subseteq L_2$, $\forall k \geq 0$, which means $L_1^* L_3 \subseteq L_2$.

We have shown that $L_2 \subseteq L_1^* L_3$ and $L_2 \supseteq L_1^* L_3$. Thus, $L_2 = L_1^* L_3$. □

Problem 3

1. We will prove by induction on i that Q_r^i is the set of all states reachable from q_0 by strings of length i.

- Basis: when $i = 0$, $Q_r^0 = \{q_0\}$ which is clearly the set of all states reachable from q_0 by the empty string ϵ whose length is 0. Because D is a DFA (deterministic finite automaton), any state of D that is different from q_0 cannot be reached from q_0 by the empty string.
- Induction: assume that Q_r^i, where $i \geq 0$, is the set of all states reachable from q_0 by strings of length i. We need to prove that Q_r^{i+1} is the set of all states reachable from q_0 by strings of length $i + 1$.
  - For any state $q \in Q_r^{i+1}$, from the definition of Q_r^{i+1}, it follows that there exist a state $p \in Q_r^i$ and a symbol $a \in A$ such that $\delta(p, a) = q$. Because $p \in Q_r^i$, we have $\delta(q_0, w) = p$ for some string w of length i. Thus, $\delta(q_0, wa) = q$ in which string wa is of length $i + 1$. Therefore, all states in Q_r^{i+1} are reachable from q_0 by some string of length $i + 1$. However, this does not prove that all states reachable from q_0 by strings of length $i + 1$ are in Q_r^{i+1}.
  - If q' is a state reachable from q_0 by a string w' such that $|w'| = i + 1$, then it must be that $w' = \hat{w}a$ for some symbol $a \in A$ and some string $\hat{w}$ of length i. Let $\hat{q} = \delta(q_0, \hat{w})$, then $\delta(\hat{q}, a) = q'$. Since $\hat{q}$ is reachable from q_0 by string $\hat{w}$ of length i, $\hat{q}$ must be in Q_r^i. From the definition of Q_r^{i+1}, it follows that $q' \in Q_r^{i+1}$. Thus, all states reachable from q_0 by strings of length $i + 1$ are in Q_r^{i+1}.

  Therefore, Q_r^{i+1} is the set of all states reachable from q_0 by strings of length $i + 1$. □

However, it is generally false that there is an index i_0 such that $Q_r^{i_0+1} = Q_r^{i_0}$. This is shown in the following counter-example. Consider the simple DFA in figure 1, with $Q = \{q_0, q_1\}$, $A = a$, $Q_0 = \{q_0\}$, and $Q_m = \emptyset$. Applying the algorithm to this DFA, we have $Q_r^0 = \{q_0\}$, $Q_r^1 = \{q_1\}$, $Q_r^2 = \{q_0\}$,

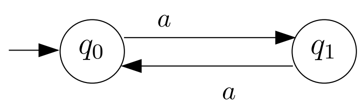

Figure 1: Counter-example of problem 3

$Q_r^3 = \{q_1\}$, ... In other words, $Q_r^i = \{q_0\}$ whenever i is even, and $Q_r^i = \{q_1\}$ whenever i is odd. Therefore, there is no index i_0 such that $Q_r^{i_0+1} = Q_r^{i_0}$.

2. Using the given algorithm, it is generally false that $Q_r^{i_0} = Q_r$ for some index i_0. In the example given in part 1 (figure 1), it is easy to see that $Q_r = Q = \{q_0, q_1\}$, however Q_r^i is either $\{q_0\}$ or $\{q_1\}$ depending on the value of i. Thus, there never exists an index i_0 such that $Q_r^{i_0} = Q_r$ for this DFA. Therefore, the statement is generally false.

3. First, we will prove by induction on i that Q_r^i is the set of all states reachable from q_0 by strings w such that $|w| \leq i$.

- Basis: when $i = 0$, $Q_r^0 = \{q_0\}$ which is the set of all states reachable from q_0 by the empty string ϵ whose length is $0 \leq i$.
- Induction: assume that Q_r^i, where $i \geq 0$, is the set of all states reachable from q_0 by strings of length less than or equal to i. For any state $q \in Q_r^{i+1}$, since

$$Q_r^{i+1} = Q_r^i \cup \{q \in Q \mid \exists p \in Q_r^i, \exists a \in \Sigma : q = \delta(p, a)\},$$

  there must exist a string w such that $|w| \leq i$ and either $\delta(q_0, w) = q$ or $\delta(q_0, wa) = q$ for some $a \in \Sigma$. Thus, all states in Q_r^{i+1} are reachable from q_0 by strings w such that $|w| \leq i+1$. On the other hand, for any state q' reachable from q_0 by a string w' such that $|w'| \leq i+1$, it must be that either $|w'| \leq i$ or, if $|w'| = i+1$, $w' = \hat{w}a$ for some symbol $a \in \Sigma$ and some string $\hat{w}$ of length i. In the latter case, let $\hat{q} = \delta(q_0, \hat{w})$, then $\delta(\hat{q}, a) = q'$ and $\hat{q} \in Q_r^i$. Thus, $q' \in Q_r^{i+1}$, which means that all states reachable from q_0 by strings of length $i+1$ or less are in Q_r^{i+1}. Therefore, Q_r^{i+1} is the set of all states reachable from q_0 by strings w with $|w| \leq i+1$.

The number of states of D is finite. For any state $q \in Q_r$, since it is reachable from q_0, there must exist a smallest integer N_q which is finite such that $\delta(q_0, w) = q$ for some string w of length N_q. Because Q_r is finite and N_q is finite for each $q \in Q_r$, $\max_{q \in Q_r} N_q$ exists and is finite. Let i_0 be this maximum value. It follows that every state q in Q_r is reachable from q_0 by a string of length i_0 or less. Thus, $Q_r^{i_0} = Q_r$ (by the above result). We also have $Q_r^{i_0} \subseteq Q_r^{i_0+1}$ (by the definition of $Q_r^{i_0+1}$) as well as $Q_r^{i_0+1} \subseteq Q_r = Q_r^{i_0}$. Hence, $Q_r^{i_0} = Q_r^{i_0+1}$.

To prove that i_0 is smallest, we assume that it is not true, i.e. there exists $j < i_0$ such that $Q_r^j = Q_r$. Then all states $q \in Q_r$ are reachable from q_0 by strings w such that $|w| \leq j < i_0$. Thus $N_q < i_0$ for all $q \in Q_r$. It follows that $\max_{q \in Q_r} N_q < i_0$ which contradicts the fact that $i_0 = \max_{q \in Q_r} N_q$. Therefore, i_0 must be smallest.

To conclude, there exists a smallest integer i_0 such that $Q_r^{i_0+1} = Q_r^{i_0} = Q_r$.

Problem 4

1. The deterministic hybrid automaton modeling the system is given in figure 2. The hybrid automaton has four discrete modes, corresponding to the four locations q_1, q_2, q_3, and q_4, which have invariants corresponding to the four quadrants. The specification, according to the definition given in reference [R1], is as follows:

- Set of locations $L = \{q_1, q_2, q_3, q_4\}$
- Continuous state space $X = \mathbb{R}^2$
- Continuous external variables space $W = \emptyset$

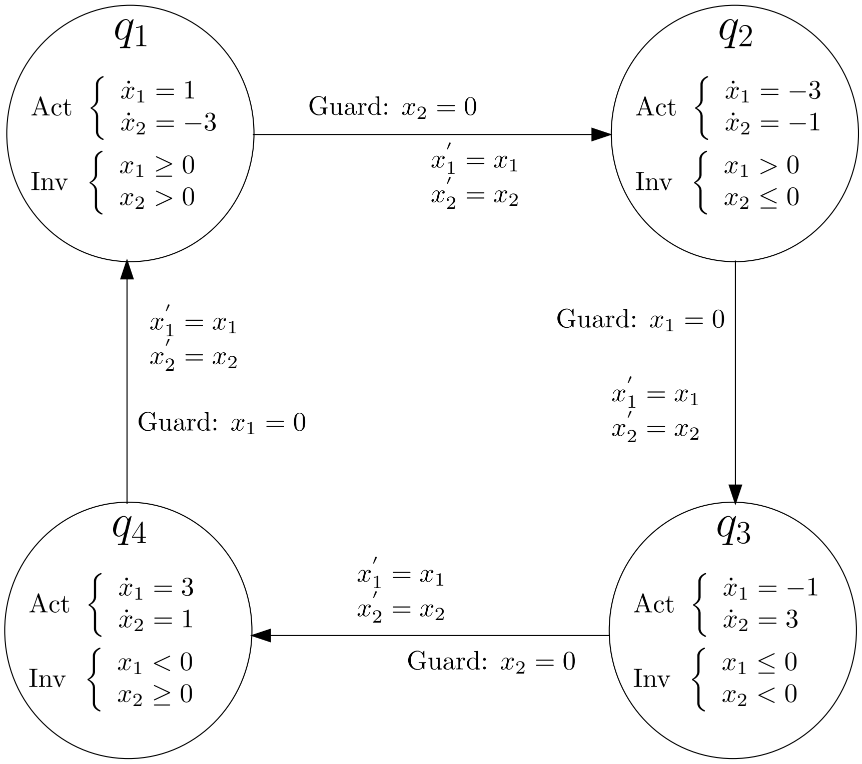

Figure 2: Hybrid Automaton modeling the system in Problem 4

- Location invariants:
  - $Inv(q_1) = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 \geq 0,\, x_2 > 0\}$
  - $Inv(q_2) = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 > 0,\, x_2 \leq 0\}$
  - $Inv(q_3) = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 \leq 0,\, x_2 < 0\}$
  - $Inv(q_4) = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 < 0,\, x_2 \geq 0\}$
- Location dynamics (activities) Act:
  - $Act(q_1) = \begin{cases} \dot{x}_1 = 1 \\ \dot{x}_2 = -3 \end{cases}$
  - $Act(q_2) = \begin{cases} \dot{x}_1 = -3 \\ \dot{x}_2 = -1 \end{cases}$
  - $Act(q_3) = \begin{cases} \dot{x}_1 = -1 \\ \dot{x}_2 = 3 \end{cases}$
  - $Act(q_4) = \begin{cases} \dot{x}_1 = 3 \\ \dot{x}_2 = 1 \end{cases}$
- Set of transitions:
  - Transition from q_1 to q_2 with $Guard_{q_1,q_2} = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_2 = 0\}$ and $Jump_{q_1,q_2} = \{(x_1, x_2, x_1^{'}, x_2^{'}) \in \mathbb{R}^4 \,|\, x_1^{'} = x_1,\, x_2^{'} = x_2\}$
  - Transition from q_2 to q_3 with $Guard_{q_2,q_3} = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 = 0\}$ and $Jump_{q_2,q_3} = \{(x_1, x_2, x_1^{'}, x_2^{'}) \in \mathbb{R}^4 \,|\, x_1^{'} = x_1,\, x_2^{'} = x_2\}$
  - Transition from q_3 to q_4 with $Guard_{q_3,q_4} = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_2 = 0\}$ and $Jump_{q_3,q_4} = \{(x_1, x_2, x_1^{'}, x_2^{'}) \in \mathbb{R}^4 \,|\, x_1^{'} = x_1,\, x_2^{'} = x_2\}$
  - Transition from q_4 to q_1 with $Guard_{q_4,q_1} = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_1 = 0\}$ and $Jump_{q_4,q_1} = \{(x_1, x_2, x_1^{'}, x_2^{'}) \in \mathbb{R}^4 \,|\, x_1^{'} = x_1,\, x_2^{'} = x_2\}$

2. For any non-zero initial continuous state $x(0) \neq 0$, the hybrid automaton starts in one of the four locations q_1, q_2, q_3, and q_4, according to the quadrant that $(x_{1,0}, x_{2,0})$ is in.

- If the hybrid automation is in location q_1, i.e. $x_1 \geq 0$ and $x_2 > 0$, x_1 keeps increasing (since $\dot{x}_1 = 1$) while x_2 keeps decreasing faster (since $\dot{x}_2 = -3$) until $x_2 = 0$. When $x_2 = 0$, the invariant is violated and the guard of the sole transition to q_2 is satisfied, thus the hybrid automaton changes to location q_2.
- If the hybrid automation is in location q_2, i.e. $x_1 > 0$ and $x_2 \leq 0$, x_2 keeps decreasing (since $\dot{x}_2 = -1$) while x_1 decreases faster (since $\dot{x}_1 = -3$) until $x_1 = 0$. When $x_1 = 0$, the invariant is violated and the guard of the sole transition to q_3 is satisfied, thus the hybrid automaton changes to location q_3.
- If the hybrid automation is in location q_3, i.e. $x_1 \leq 0$ and $x_2 < 0$, x_1 keeps decreasing (since $\dot{x}_1 = -1$) while x_2 increases faster (since $\dot{x}_2 = 3$) until $x_2 = 0$. When $x_2 = 0$, the invariant is violated and the guard of the sole transition to q_4 is satisfied, thus the hybrid automaton changes to location q_4.
- If the hybrid automation is in location q_4, i.e. $x_1 < 0$ and $x_2 \geq 0$, x_2 keeps increasing (since $\dot{x}_2 = 1$) while x_1 increases faster (since $\dot{x}_1 = 3$) until $x_1 = 0$. When $x_1 = 0$, the invariant is violated and the guard of the sole transition to q_1 is satisfied, thus the hybrid automaton changes to location q_1.

From above, we can see that the hybrid automaton keeps switching repeatedly between the four locations, without making (x_1, x_2) reach the origin.

On the other hand, it is easy to see that in each location we have $\frac{\mathrm{d}}{\mathrm{d}t}(|x_1(t)| + |x_2(t)|) = -2$. Therefore, the sum $|x_1(t)| + |x_2(t)|$ decreases with time (in other words, (x_1, x_2) goes to $(0, 0)$) and (x_1, x_2) reaches the origin after $\frac{1}{2}(|x_{1,0}| + |x_{2,0}|)$ units of time. However, the continuous state cannot arrive at the origin without going through an infinite number of transitions between the four locations q_1, q_2, q_3, and q_4 (shown above).

It follows that the system has Zeno execution for every non-zero initial state. The Zeno time is $\frac{1}{2}(|x_{1,0}| + |x_{2,0}|)$.

Problem 5

1. This system has a livelock whenever x_1 reaches 0. It is because when $x_1 = 0$, $\operatorname{sgn}(x_1)$ is undefined, thus x_1 may become either positive ($x_1 > 0$) or negative ($x_1 < 0$). However, since $\dot{x}_1 = -\operatorname{sgn}(x_1)$, variable x_1 will return to 0 immediately. This is repeated again and again, and the system switches infinitely between the two modes: the mode when $x > 0$ and the mode when $x < 0$. Thus, the system has a livelock.

2. Using the forward Euler method to approximate the derivatives of x_1 and x_2 with respect to time, we have

$$\frac{x_{1,k+1} - x_{1,k}}{h} = -\operatorname{sgn}(x_{1,k})$$
$$\frac{x_{2,k+1} - x_{2,k}}{h} = -x_{2,k}$$

which leads to

$$x_{1,k+1} = x_{1,k} - h\,\text{sgn}(x_{1,k})$$
$$x_{2,k+1} = x_{2,k} - hx_{2,k}$$

With initial condition $(x_{1,0}, x_{2,0}) = (1, 1)$, we can simulate the execution of the system using the formulae above with $k = 1, 2, 3, \ldots$, for time $0 \leq t = k.h \leq 5$. For three different values of time step $h = 0.1$, 0.05, 0.01, we have three simulations. Their results are given in figure 3. The upper plot shows the values of $x_1(t)$ and $x_2(t)$ of all three simulations. The lower plot shows the graph of (x_1, x_2) in the state space plane. In the graphs, we can see the repeated switches of the system between the two modes: $x > 0$ and $x < 0$, which illustrate the livelock of the system.

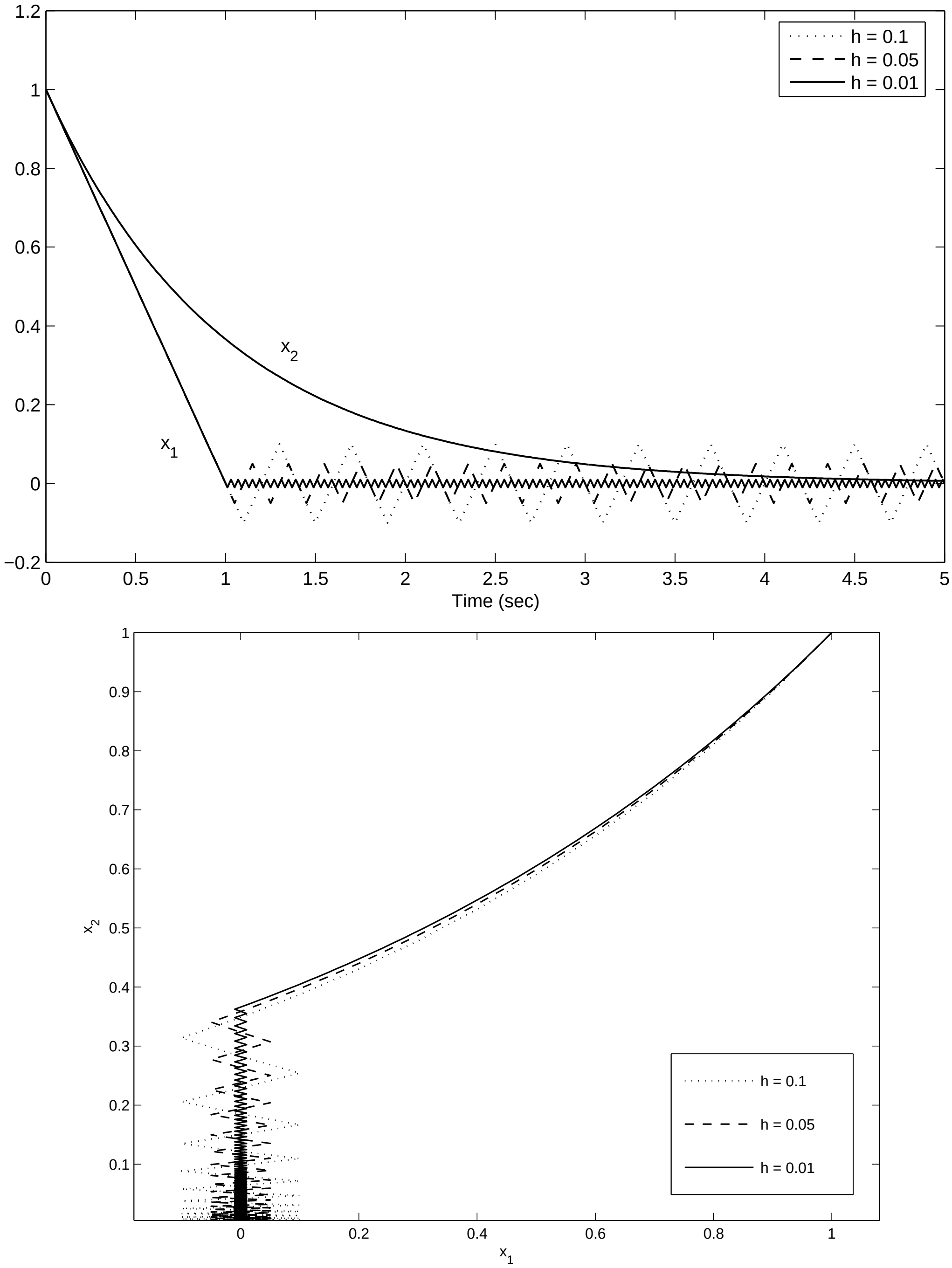

Figure 3: Simulation results with $h = 0.1$ (dotted line), $h = 0.05$ (dashed line), and $h = 0.01$ (solid line)

3. With the new definition of sgn(.) we have

- The differential equation $\dot{x}_2 = -x_2$ gives the solution $x_2(t) = x_{2,0}e^{-t}$. With $x_{2,0} = 1$, we have $x_2(t) = e^{-t}$, $t \geq 0$
- With $x_{1,0} = 1 > 0$, we have the differential equation $\dot{x}_1 = -\operatorname{sgn}(x_1) = -1$, which gives the solution $x_1(t) = 1 - t$ for $t \geq 0$ and while $x_1 > 0$. At time $t = 1$, x_1 is 0 and, since $\operatorname{sgn}(0) = 0$, the differential equation for x_1 becomes $\dot{x}_1 = 0$. Thus, after time instant $t = 1$, the value of x_1 does not change and is 0. Mathematically, we have:

$$x_1(t) = \begin{cases} 1 - t & \text{if } 0 \leq t < 1 \\ 0 & \text{if } t \geq 1 \end{cases}$$

In this solution, we do not see the repeated switches of the system between the two modes, $x_1 > 0$ and $x_1 < 0$, as in the results of the previous part. It is because we introduced a new mode (the sliding mode) to the system, corresponding to $x_1 = 0$, by defining the value of sgn(0) to be 0. Therefore, the new system does not have a livelock as does the original system. If we plot the graph of (x_1, x_2) of the new system, we will have the result as in figure 4. As we can see, the main difference between the plots in the previous part and this plot is that there are no repeated switches between $x < 0$ and $x > 0$ in this plot. The vertical line represents the sliding mode of the system.

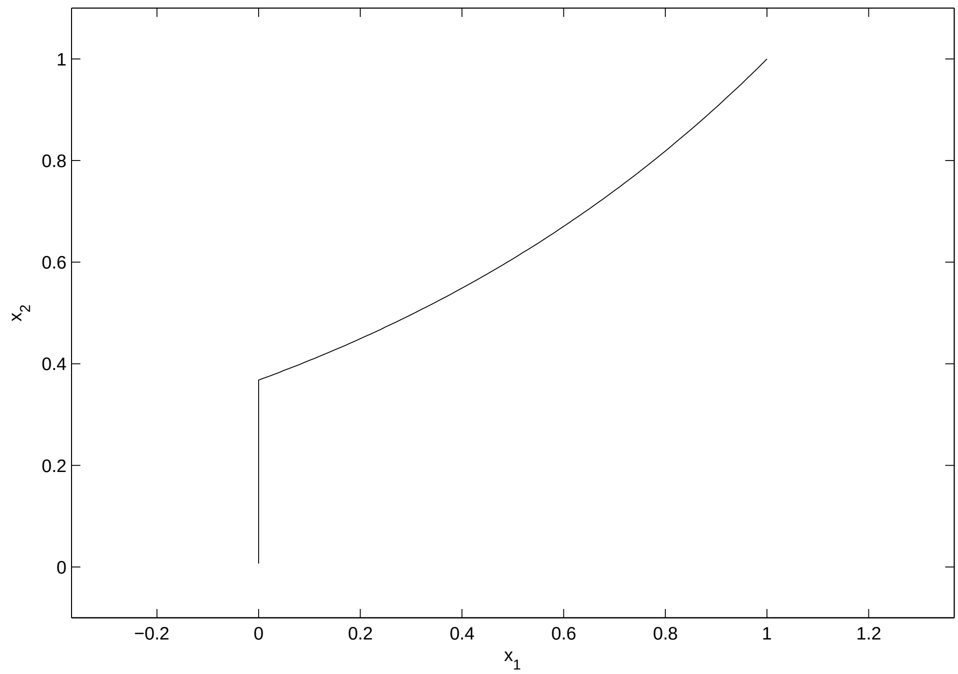

Figure 4: Graph of (x_1, x_2) of the new system with the sliding mode