

非線性控制
Nonlinear Control

第七章作業
逆向步進控制

學號：P46104285

研究生：楊亞勳

授課教授：楊憲東

Department of Aeronautics and Astronautics

National Cheng Kung University

Tainan, Taiwan, ROC

中華民國 111 年 12 月 10 日

目錄

Question 1	1
Question 2	7
Question 3	11
Question 4	13
Question 5	14
MATLAB Code	19

考慮二階非線性系統

$$\dot{x}_1 = 2x_1 + x_1^2 + x_1x_2 \quad (1a)$$

$$\dot{x}_2 = 3x_1 + (1 + x_2^2)u \quad (1b)$$

提示，如果依據(7.4.2)式~(7.4.7)式無法設計出所需要的控制器，試著先進行座標平移 $y_1 = x_1$ ， $y_2 = x_2 + 2$ ，然後在 (y_1, y_2) ，的座標下進行逆向步進控制及 MATLAB 模擬，討論為甚麼需要座標平移。

Question 1

列出逆向步進控制的設計步驟，參考(7.4.2)式~(7.4.7)式，設計控制律 $u(x_1, x_2)$ ，使得控制後的系統(亦即閉迴路系統)相對於原點為漸進穩定。

Answer

根據(1a)和(1b)之二階非線性系統，可知此系統為一二層結構，根據課本(7.4.2)~(7.4.7)式，先寫出此系統之一般式

$$\dot{x} = f_0(x) + g_0(x)\xi_1 \quad (2a)$$

$$\dot{\xi}_1 = f_1(x, \xi_1) + g_1(x, \xi_1)u \quad (2b)$$

此外(2)的形式還可透過以下控制訊號

$$u = \frac{1}{g_1(x, \xi_1)} [u_1 - f_1(x, \xi_1)] \quad (3)$$

轉換成標準形式

$$\dot{x} = f_0(x) + g_0(x)\xi_1 \quad (4a)$$

$$\dot{\xi}_1 = u_1 \quad (4b)$$

此標準形式之控制律可由以下式子表示

$$u_1 = \phi_2(x, \xi_1) = \frac{\partial \phi_1}{\partial x} [f_0(x) + g_0(x)\xi_1] - \frac{\partial V_1}{\partial x} g_0(x) - k[\xi_1 - \phi_1(x)] \quad (5)$$

此控制律可使(4)式之系統為漸進穩定，也就是狀態變數 $[x, \xi_1]$ 趨近於原點(平衡點)，而與(5)對應之 Lyapunov 函數如下

$$V_2(x, \xi_1) = V_1(x) + \frac{(\xi_1 - \phi_1(x))^2}{2} \quad (6)$$

其中 $\phi_1(x)$ 和 $V_1(x)$ 是內層子系統(4a)的控制律和 Lyapunov 函數。最後將(5)帶入(3)式即可得到原系統(2)之穩定控制律

$$u = \frac{1}{g_1(x, \xi_1)} \left[\frac{\partial \phi_1}{\partial x} [f_0(x) + g_0(x)\xi_1] - \frac{\partial V_1}{\partial x} g_0(x) - k[\xi_1 - \phi_1(x)] - f_1(x, \xi_1) \right] \quad (7)$$

(2)~(7)式即為二層結構非線性系統的一般形式逆向步進控制。逆向步進控制的發展是為了解決回授線性化控制所面臨的問題，包括系統參數測量誤差造成的不完全線性化、捨棄非線性快速收斂性質和控制訊號可能超過可實現範圍等等。而逆向步進控制的核心概念為利用 Lyapunov 函數來設計一控制器，使非線性系統之所有狀態變數隨著時間推移收斂至原點達到漸進穩定。但是和前面課本章節所遇到的低階純量系統相比，實際上的非線性系統可能為高階多變數系統，這讓找尋一個能代表所有狀態變數的 Lyapunov 函數變得非常困難。而逆向步進控制利用層狀結構的方法將系統拆分為多層結構，一層一層的找尋 Lyapunov 函數和相對應知控制律，待每一層的 Lyapunov 函數和控制律皆求得後，即可得到全系統的控制律和相對應知 Lyapunov 函數，此即逆向步進控制(Backstepping Control)。在開始進行逆向步進控制器設計前，我們先看看是否需要進行提示提及的座標平移。會需要進行座標平移，首先猜測是因為此系統的平衡點並不在原點(0,0)上。為了檢驗平衡點是否在原點上，首先考慮式(1a)

$$\dot{x}_1 = 2x_1 + x_1^2 + x_1x_2 \quad (1a)$$

此即系統的內層迴路系統，若將 x_2 做為控制輸入，並設計控制律 $x_2 = \phi(x_1)$ ，使得內層閉迴路子系統為漸進穩定，且可由以下式子表示

$$\dot{x}_1 = 2x_1 + x_1^2 + x_1\phi(x_1) \quad (8)$$

設計控制律 $x_2 = \phi(x_1)$ 使得內層子系統為漸進穩定。亦即找到 Lyapunov 函數 $V_1(x_1) > 0$ 滿足

$$\dot{V}_1(x_1) = \frac{\partial V_1}{\partial x_1} [f_0(x_1) + g_0(x_1)\phi(x_1)] \leq -V_a(x_1) < 0 \quad (9)$$

其中 $V_a(x_1)$ 為正定函數，所得到之控制律 $x_2 = \phi(x_1)$ 必須確保閉迴路子系統(16a)為漸進穩定。對於內層子系統(1a)，選擇 Lyapunov 函數

$$V_1(x_1) = \frac{1}{2}x_1^2 \quad (10)$$

將(10)對系統狀態做一次導數

$$\begin{aligned} \dot{V}_1(x_1) &= x_1\dot{x}_1 = x_1[2x_1 + x_1^2 + x_1\phi(x_1)] \\ &= 2x_1^2 + x_1^3 + x_1^2\phi(x_1) \end{aligned} \quad (11)$$

將(9)和(11)結合，可以得到

$$\dot{V}_1(x_1) = 2x_1^2 + x_1^3 + x_1^2\phi(x_1) \leq -V_a(x_1) < 0 \quad (12)$$

我們在此可選擇先指定 \dot{V} ，再決定控制律 $x_2 = \phi(x_1)$ 。根據式(12)，若是要符合 $V_a(x_1)$ 為正定函數的條件，又最低次方形式為 x_1^3 ，故選擇 $V_a(x_1) = x_1^4 > 0, \forall x_1 \neq 0$ 。將此正定函數代回(12)中，可以得到

$$\phi(x_1) = -2 - x_1 - x_1^2 \quad (13)$$

若將式(13)代回(8)式，則可得到

$$\dot{x}_1 = -x_1^3 \quad (14)$$

由式子(14)可知此系統加上控制律 $x_2 = \phi(x_1)$ 後仍然是一非線性系統，平衡點在 $x_1 = 0$ 。此平衡點為穩定的平衡點，因當 $x_1 > 0$ 時， $\dot{x}_1 < 0$ ；當 $x_1 < 0$ 時， $\dot{x}_1 > 0$ 。也因此系統只有一平衡點，固為全域穩定。此時再看

$$\dot{x}_2 = 3x_1 + (1 + x_2^2)u \quad (1b)$$

我們希望能找到一個控制律 u ，能夠使(1b)的狀態變數 $x_2 \rightarrow \phi(x_1)$ ，這樣能使系統的最終狀態達到 $x_1 = 0$ 。但是觀察內層控制律，當 $x_1 = 0$ 時， $\phi(x_1) = -2$ ，又因 $x_2 \rightarrow \phi(x_1)$ ，故可知此系統最終的平衡點為 $(x_1, x_2) = (0, -2)$ ，並非題目所要求的原點。故此時需要對系統做座標平移 $y_1 = x_1$ ， $y_2 = x_2 + 2$ ，得到新的系統

$$\dot{y}_1 = y_1^2 + y_1 y_2 \quad (15a)$$

$$\dot{y}_2 = 3y_1 + (1 + (y_2 - 2)^2)u \quad (15b)$$

此時的目標為設計控制律 $u(y_1, y_2)$ ，使得控制後的系統(亦即閉迴路系統)相對於原點為漸進穩定。將(15a)、(15b)和(2a)、(2b)做對比後可以得到

$$f_0(y_1) = y_1^2 \quad (16)$$

$$g_0(y_1) = y_1 \quad (17)$$

$$f_1(y_1, y_2) = 3y_1 \quad (18)$$

$$g_1(y_1, y_2) = 1 + (y_2 - 2)^2 \quad (19)$$

對於上述系統，當 $f_1(y_1, y_2) \neq 0$ 時，根據(3)，可以透過控制訊號轉換得到

$$u = \frac{1}{1 + (y_2 - 2)^2} [u_1 - 3y_1] \quad (20)$$

若是將(20)代回(15a)、(15b)可以得到

$$\dot{y}_1 = y_1^2 + y_1 y_2 \quad (21a)$$

$$\dot{y}_2 = u_1 \quad (21b)$$

接著依照以下四個步驟進行逆向步進控制器設計：

(1) 設計內層結構的控制律

設計控制律 $y_2 = \phi(y_1)$ 使得內層子系統為漸進穩定。亦即找到 Lyapunov 函數 $V_1(y_1) > 0$ 滿足

$$\dot{V}_1(y_1) = \frac{\partial V_1}{\partial y_1} [f_0(y_1) + g_0(y_1)\phi(y_1)] \leq -V_a(y_1) < 0 \quad (22)$$

其中 $V_a(y_1)$ 為正定函數，所得到之控制律 $y_2 = \phi(y_1)$ 必須確保閉迴路子系統(21a)為漸進穩定。對於內層子系統(21a)，選擇 Lyapunov 函數

$$V_1(y_1) = \frac{1}{2} y_1^2 \quad (23)$$

將(23)對系統狀態做一次導數

$$\begin{aligned} \dot{V}_1(y_1) &= y_1 \dot{y}_1 = y_1 [y_1^2 + y_1 \phi(y_1)] \\ &= y_1^3 + y_1^2 \phi(y_1) \end{aligned} \quad (24)$$

將(24)和(22)結合，可以得到

$$\dot{V}_1(y_1) = y_1^3 + y_1^2 \phi(y_1) \leq -V_a(y_1) < 0 \quad (25)$$

我們在此可選擇先指定 \dot{V} ，再決定控制律 $y_2 = \phi(y_1)$ 。根據式(25)，若是要符合 $V_a(y_1)$ 為正定函數的條件，又最低次方形式為 y_1^3 ，故選擇 $V_a(y_1) = y_1^4 > 0, \forall y_1 \neq 0$ 。將此正定函數代回(25)中，可以得到

$$\phi(y_1) = -y_1 - y_1^2 \quad (26)$$

將(26)代回(24)並且整理如下

$$\dot{V}_1(y_1) = -y_1^4 \leq 0 \quad (27)$$

根據(27)我們可以知道，控制律(26)的設計能夠讓狀態 y_1 隨著時間推進收斂到 0。也就是符合漸進穩定的需求。此時內層控制律 $y_2 = \phi(y_1)$ 和 Lyapunov 函數 $V_1(y_1)$ 已確立。

(2) 建立外層結構運動方程式

將(21a)和(21b)改寫成以下形式

$$\dot{y}_1 = y_1^2 + y_1 y_2 + y_1 (y_2 - \phi(y_1)) \quad (28a)$$

$$\dot{y}_2 = u_1 \quad (28b)$$

由(28)式觀察可知，變數 y_2 是控制訊號 u_1 的積分結果，而變數 y_2 又是(21a)子系統的控制訊號，這代表進入(28a)子系統的控制訊號變成了 $y_2 - \phi(y_1)$ 。

(3) 定義外層結構的控制訊號

令

$$z = y_2 - \phi(y_1) \quad (29)$$

可以得到

$$\dot{z} = \dot{y}_2 - \dot{\phi}(y_1) = u_1 - \dot{\phi}(y_1) \quad (30)$$

此時可定義新的控制訊號

$$v = u_1 - \dot{\phi}(y_1) \quad (31)$$

結合(28a)式、(30)式和(31)式，可以得到變換後的系統方程式

$$\dot{y}_1 = y_1^2 + y_1 (-y_1 - y_1^2) + y_1 z \quad (32a)$$

$$\dot{z} = v \quad (32b)$$

經過轉換後，子系統的控制訊號為 z ，全系統的控制訊號為 v 。

(4) 決定外層結構控制律

當 $z=0$ 時，根據以上設計內層系統

$$\dot{y}_1 = y_1^2 + y_1 \phi(y_1) \quad (33)$$

相對於原點為漸進穩定，接下來設計全系統控制訊號 v 使得 z 趨近於 0。設 $V(y_1, z)$ 為全系統之 Lyapunov 函數，將其取以下形式

$$V(y_1, z) = V_1(y_1) + \frac{z^2}{2} = \frac{1}{2} y_1^2 + \frac{z^2}{2} \quad (34)$$

若將(34)對狀態變數做一次導數

$$\begin{aligned} \dot{V}(y_1, z) &= y_1 \dot{y}_1 + z \dot{z} \\ &= y_1 (y_1^2 + y_1 (-y_1 - y_1^2) + y_1 z) + z v \\ &= y_1^3 - y_1^3 - y_1^4 + y_1^2 z + z v \\ &= -y_1^4 + y_1^2 z + z v \end{aligned} \quad (35)$$

根據課本，為了使 $\dot{V}(y_1, z) < 0$ ，可選控制律如下

$$v = -\frac{\partial V_1}{\partial y_1} g_0(y_1) - kz = -y_1^2 - kz, \quad k > 0 \quad (36)$$

將(36)式代入(35)式中，得到

$$\dot{V}(y_1, z) = -y_1^4 - kz^2 < 0 \quad (37)$$

(37)式說明當 $y_1 \neq 0$ 、 $z \neq 0$ 時， $\dot{V}(y_1, z) < 0$ 。此條件保證了狀態變數 $[y_1 \ y_2]^T$ 會隨著時間的推移趨近於平衡點(原點)，也就是全系統的漸進穩定。接著將(20)式、(30)式、(32b)式和(36)式，可得到原系統之控制律 u 如下

$$\begin{aligned} u(y_1, y_2) &= \frac{1}{1+(y_2-2)^2} [u_1 - 3y_1] \\ &= \frac{1}{1+(y_2-2)^2} [v + \dot{\phi}(y_1) - 3y_1] \\ &= \frac{1}{1+(y_2-2)^2} \left[-\frac{\partial V_1}{\partial y_1} g_0(y_1) + \dot{\phi}(y_1) - 3y_1 \right] \\ &= \frac{1}{1+(y_2-2)^2} \left[-\frac{\partial V_1}{\partial y_1} g_0(y_1) - kz + \frac{\partial \phi(y_1)}{\partial y_1} (f_0(y_1) + g_0(y_1) y_2) - 3y_1 \right] \\ &= \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1 y_2) - y_1^2 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \end{aligned} \quad (38)$$

此時內層子系統和全系統的控制律和相應之 Lyapunov 函數已確立。回顧原系統

$$\dot{y}_1 = y_1^2 + y_1 y_2 \quad (15a)$$

$$\dot{y}_2 = 3y_1 + (1 + (y_2 - 2)^2)u \quad (15b)$$

其中

$$y_2 = \phi(y_1) = -y_1 - y_1^2 \quad (26)$$

$$u(y_1, y_2) = \frac{1}{1 + (y_2 - 2)^2} \left[(-1 - 2y_1)(y_1^2 + y_1 y_2) - y_1^2 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \quad (38)$$

透過以上的設計流程，我們可以得到系統(15)的逆向步進控制器設計結果。透過控制輸入 u ，使得狀態 y_2 會趨於子系統控制輸入 ϕ ，而子系統控制輸入 ϕ 又使得內層系統狀態 y_1 趨於 0，當內層系統狀態 y_1 趨於 0 時，透過(26)式可以知道外層系統狀態 y_2 也會趨近於 0，故此系統最終會收斂至原點 $(y_1, y_2) = (0, 0)$ 。若是比照至原系統，也就是座標平移前的系統，則會收斂至 $(x_1, x_2) = (0, -1)$ 。

Question 2

將所得到的 $u(x_1, x_2)$ 代回(1)式，進行閉迴路的 MATLAB 模擬。選定不同的初始點，畫出相平面軌跡，驗證軌跡相對於原點的漸進穩定。

Answer

呈上題所得到的非線性系統及設計結果

$$\dot{y}_1 = y_1^2 + y_1 y_2 \quad (15a)$$

$$\dot{y}_2 = 3y_1 + (1 + (y_2 - 2)^2)u \quad (15b)$$

其中

$$y_2 = \phi(y_1) = -y_1 - y_1^2 \quad (26)$$

$$u(y_1, y_2) = \frac{1}{1 + (y_2 - 2)^2} \left[(-1 - 2y_1)(y_1^2 + y_1 y_2) - y_1^2 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \quad (38)$$

針對此系統，選擇 $k=2$ ，且選定下表所列的 10 個初始點來繪製相平面軌跡圖。

表 1、選定之 10 個初始點

$(x_1, x_2) = (1, 1)$	$(x_1, x_2) = (0.5, 3)$	$(x_1, x_2) = (-1, -3)$	$(x_1, x_2) = (-2, -4)$	$(x_1, x_2) = (0.7, -2)$
$(x_1, x_2) = (2, -3)$	$(x_1, x_2) = (-4, 1)$	$(x_1, x_2) = (-0.9, 2)$	$(x_1, x_2) = (2.3, 2.7)$	$(x_1, x_2) = (4, -3)$

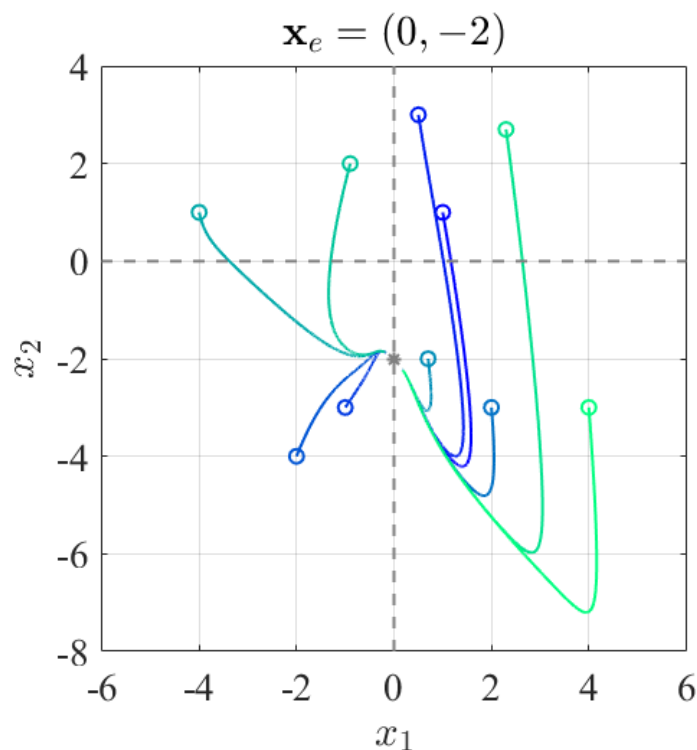


圖 1、座標平移前之系統相平面軌跡圖

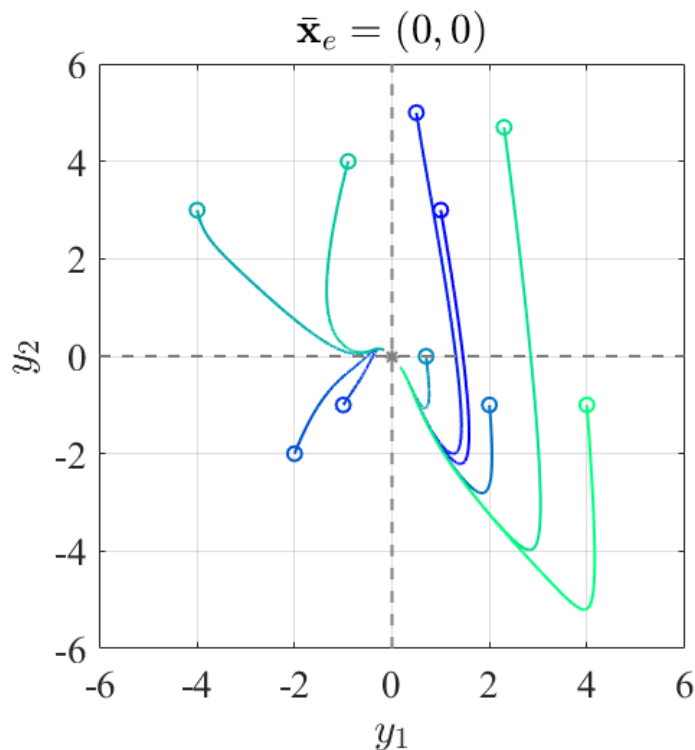


圖 2、座標平移後之系統相平面軌跡圖

結果如圖 1 和圖 2 所示。此 MATLAB 模擬使用之時間長度為 10 秒，每點間隔 0.001 秒。圖 1 繪製出座標平移前，在使用和座標平移後相同控制器設計的情況下，系統的像平面軌跡圖，由圖中可以看到所選定的 10 點分布在 4 個象限，且每一點都漸進收斂至 $(x_1, x_2) = (0, -2)$ 這點，這和我們在第一小題的式(13)的推論相同，在座標平移前，當系統狀態 $x_1 \rightarrow 0$ 時， $x_1 = 0$ 時， $x_2 \rightarrow \phi(x_1) = \phi(0) = -2$ ，而非趨近於原點。而圖 2 則繪製出座標平移後的相平面軌跡圖，和座標平移前相同，所選定的 10 點分布在 4 個象限，每一點都漸進收斂至 $(y_1, y_2) = (0, 0)$ ，也就是原點，這驗證了軌跡相對於原點的漸進穩定。

但是我們也可以由圖中觀察到一個現象，此系統的模擬時長為 10 秒，但當系統運行 10 秒後，不論是在圖 1 或是圖 2，雖然軌跡有收斂至 $(x_1, x_2) = (0, -2)$ 和 $(y_1, y_2) = (0, 0)$ 的趨勢，但是並未完全收斂至該點，圖 3 為放大圖 2 原點附近的相平面軌跡圖，根據圖 3 可以更清楚的看到，系統運作 10 秒後，相平面軌跡還是沒有收斂至原點的位置。為了更清楚的檢視這個現象，我們可以畫出兩個系統的兩個狀態變數對時間的響應圖，此次模擬時長增加至 20 秒，並觀察系統狀態變數的收斂現象。結果如圖 4 ~ 圖 7。

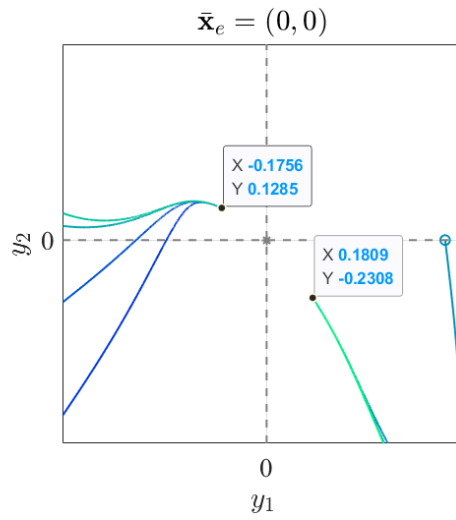


圖 3、座標平移後，模擬 10 秒原點附近系統相平面軌跡圖

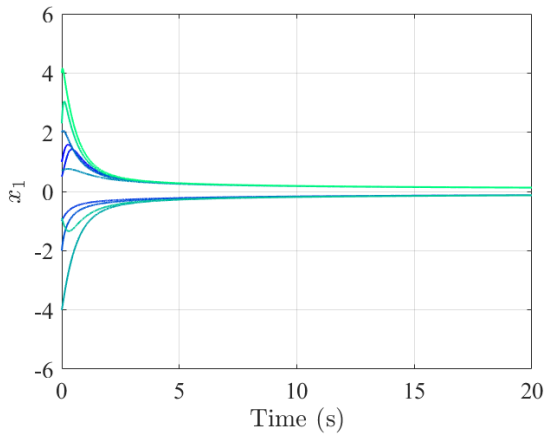


圖 4、座標平移前 x_1 時間響應

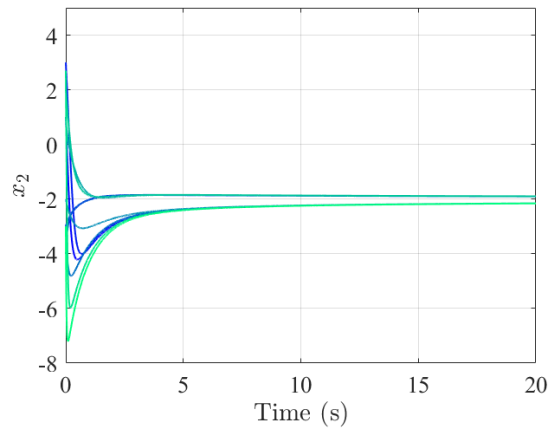


圖 5、座標平移前 x_2 時間響應

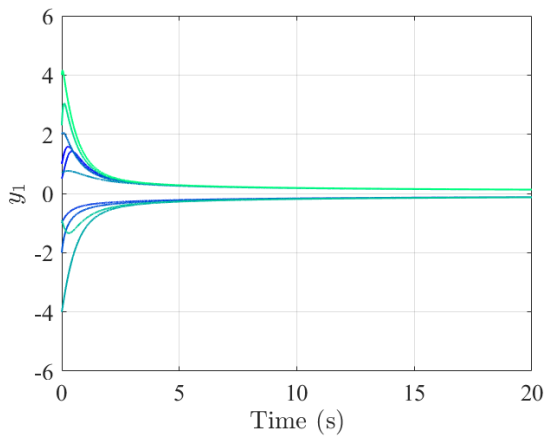


圖 6、座標平移前 y_1 時間響應

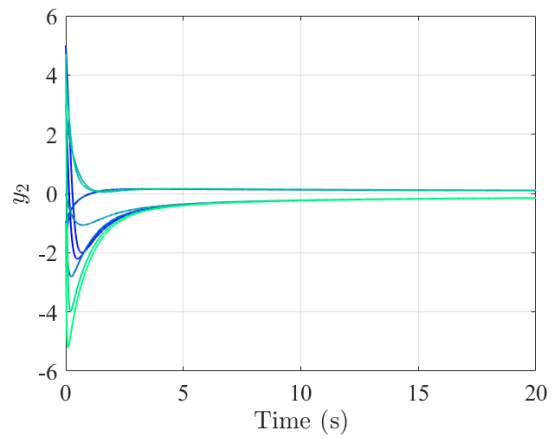


圖 7、座標平移前 y_2 時間響應

由圖 4~7 觀察可以看到，不論是座標平移前或是後的狀態變數，其值皆會”靠近”我們預想中的值，也就是 $x_1 \rightarrow 0$ 、 $x_2 \rightarrow -2$ 、 $y_1 \rightarrow 0$ 和 $y_2 \rightarrow 0$ ，但即使時長增加，這些值似乎不會達到完全的收斂，這是由極慢的系統收斂速度導致。這個現象可以用下面兩個式子解釋

$$\dot{x}_1 = -x_1^3 \quad (39)$$

$$\dot{y}_1 = -y_1^3 \quad (40)$$

(39)和(40)是由座標平移前和後，利用所設計的內層子系統控制器 ϕ 代回內層子系統的結果。觀察這兩個式子，可以看到當 $x_1 \rightarrow 0$ 、 $y_1 \rightarrow 0$ 時， $\dot{x}_1 \rightarrow 0$ 、 $\dot{y}_1 \rightarrow 0$ ，代表其狀態變數越接近平衡點，其變化量也會趨近於 0，這就是導致極慢收斂速度的原因。要解決此問題，可以嘗試其他座標平移或是控制器設計的選擇。但是在學理上，座標平移後的系統相平面軌跡相對於原點的漸進穩定。

Question 3

擴大初始點的分布範圍，從 MATLAB 的相平面軌跡圖，判斷所得到的閉迴路系統是區域穩定還是全域穩定？

Answer

在第二小題中，我們選擇的初始點皆位於正負 4 以內，為了驗證此閉迴路系統是否為全域穩定，我們可以擴大初始點選擇的範圍。在此小題中，我們選擇每一組初始點的策略為固定與原點的距離，也就是半徑，選取 16 點，以 x 軸 ($\theta = 0$) 為起點，以角度間隔 $2\pi/16$ 的方式平均分布在 4 個象限中。若是此組初始點每一點產生的相平面軌跡相對於原點的漸進穩定，就擴大半徑，選取新的一組 16 點初始點。這個方法能夠避免直接選取初始點造成在平面上象限分布不均勻的現象，而且可以用改變半徑的方式來擴大或是縮小初始點範圍。在此我們選取五個半徑，分別為 $r_1 = 0.2$ 、 $r_2 = 3$ 、 $r_3 = 20$ 、 $r_4 = 150$ 和 $r_5 = 1000$ 。模擬時長皆為 50 秒，使用的系統為座標平移後的系統，結果如圖 8~12。

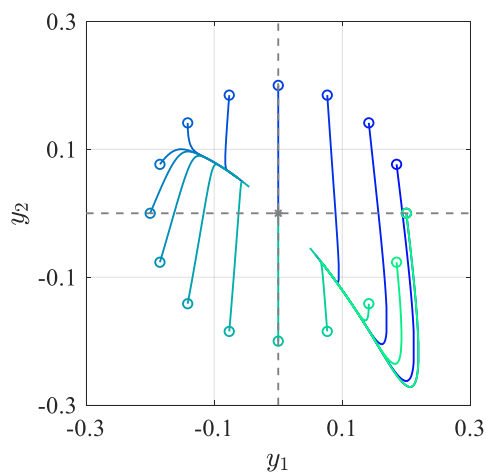


圖 8、初始點 $r_1 = 0.2$ 之相平面軌跡

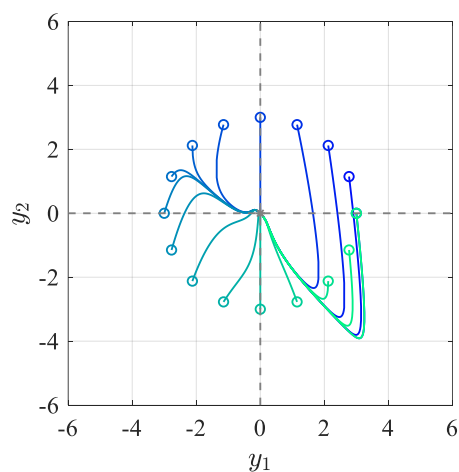


圖 9、初始點 $r_2 = 3$ 之相平面軌跡

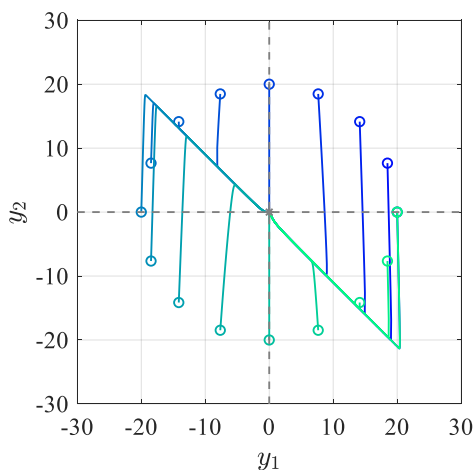


圖 10、初始點 $r_3 = 20$ 之相平面軌跡

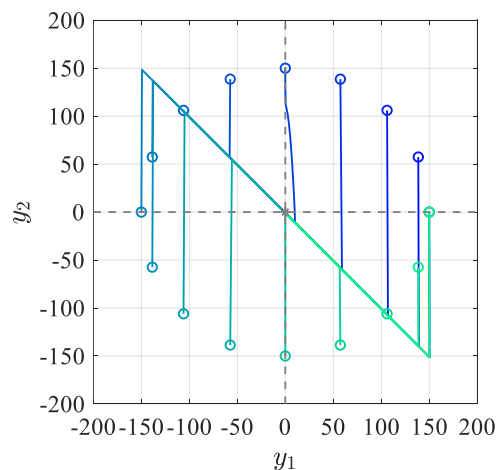


圖 11、初始點 $r_4 = 150$ 之相平面軌跡

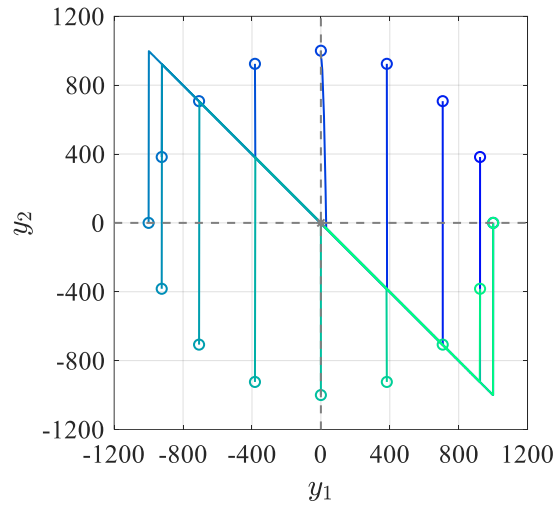


圖 12、初始點 $r_5 = 1000$ 之相平面軌跡

由圖 8~12 中觀察可知，初始點到原點的距離，不論是小至 0.2，或大至 1000，這些初始點皆可以漸進穩定的方式趨近於原點，但是在圖 8 中，極慢的收斂速度的問題變得非常明顯，可以看到即使模擬時長已經增加至 50 秒，還是未完全收斂至 0。在模擬中還發現了一個問題，當半徑增加後，原先的 RK4 數值求解器會使相平面軌跡發散至非常大的數，但若將 RK4 改為 ode45 這種可以變換積分步長和 Runge-Kutta 階數的數值求解器，即可順利收斂。成上所述，我們選擇的初始點位置平均分布在四個象限，大小從 0.2~1000 皆可完成收斂，故在此推論此系統為全域穩定。

Question 4

檢視問題 1 所得的 Lyapunov 函數 $V(x_1, x_2)$ ，從學理上判斷它是保證區域穩定還是全域穩定? 與 MATLAB 的模擬結果是否相符?

Answer

全域穩定的條件為不管 $x(0)$ 在何處(可為任一點)，恆有 $x(t) \rightarrow 0$ 時，則稱平衡點 $x=0$ 為全域穩定。若是要從 Lyapunov 函數來推論系統是否為全域穩定，則需要使用全域穩定定理(又稱 Barbashin-Krasovskii 定理)，描述如下

設 $x=0$ 為 $\dot{x}=f(x)$ 之平衡點，且 $V(x):R^n \rightarrow R$ 是一連續可微函數，且滿足下列三條件:

$$(1) V(0)=0 \text{ 且 } V(x) > 0, \forall x \neq 0$$

$$(2) \|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$$

$$(3) \dot{V}(x) < 0, \forall x \neq 0$$

則 $x=0$ 為全域漸進穩定。

有了全域穩定定理，再來看由第一小題所求出的 Lyapunov 函數如下

$$V(y_1, z) = V_1(y_1) + \frac{z^2}{2} = \frac{1}{2}y_1^2 + \frac{z^2}{2} \quad (41)$$

由(41)可以看出，此題之 Lyapunov 函數內的項皆為平方項，故符合全域穩定定理的(1)條件。接著來檢視第二個條件，在這之前，可以利用 $z = y_2 - \phi(y_1)$ 和 $\phi(y_1) = -y_1 - y_1^2$ 將(41)轉為

$$\begin{aligned} V(y_1, y_2) &= \frac{1}{2}y_1^2 + \frac{1}{2}(y_2 - \phi(y_1))^2 \\ &= \frac{1}{2}\left(y_1^2 + (y_2 + y_1 + y_1^2)^2\right) \\ &= \frac{1}{2}\left(y_1^4 + 2y_1^3 + 2y_1^2 + y_2^2 + 2y_1y_2 + 2y_1^2y_2\right) \end{aligned} \quad (42)$$

由(42)可以看到，當 $\|(y_1, y_2)\| \rightarrow (\infty, \infty)$ 時， $V(y_1, y_2) \rightarrow \infty$ ，故此題之 Lyapunov 函數符合第二個全域穩定條件。接著需要檢驗最後一個條件，但此條件在第一小題設計全系統控制器時已經將此條件納入考量，如下

$$\dot{V}(y_1, z) = -y_1^4 - kz^2 < 0 \quad (43)$$

至此，已經驗證此系統之 Lyapunov 函數符合全域穩定定理之條件，從學理上判斷它是全域穩定，與 MATLAB 的模擬結果相符。

Question 5

檢查問題 1 所列出的設計步驟，判斷所得到的控制律 $u(x_1, x_2)$ 是否為唯一？如果不是，嘗試求得另一個可以保證漸進穩定的控制律 $u'(x_1, x_2)$ ，並比較 $u'(x_1, x_2)$ 與 $u(x_1, x_2)$ 的模擬結果。

Answer

觀察第一小題的四個設計步驟，可以發現有三個地方可以改變控制律 $u(x_1, x_2)$ 的設計結果。第一個是內層子系統 Lyapunov 函數 $V_1(y_1)$ 的選擇；第二個是要使 $\dot{V}_1(y_1) < 0$ 的正定函數 $V_a(y_1)$ 的選擇；第三個是控制律 $v = -y_1^2 - kz$ 中 k 的選擇。此題會依照這三個選擇依序改變，並觀察對系統的影響。此題繼續採用座標平移後的系統

$$\begin{aligned}\dot{y}_1 &= y_1^2 + y_1 y_2 \\ \dot{y}_2 &= 3y_1 + (1 + (y_2 - 2)^2)u\end{aligned}\tag{15}$$

控制訊號轉換

$$f_0(y_1) = y_1^2\tag{16}$$

$$g_0(y_1) = y_1\tag{17}$$

$$f_1(y_1, y_2) = 3y_1\tag{18}$$

$$g_1(y_1, y_2) = 1 + (y_2 - 2)^2\tag{19}$$

$$u = \frac{1}{1 + (y_2 - 2)^2} [u_1 - 3y_1]\tag{20}$$

得到

$$\begin{aligned}\dot{y}_1 &= y_1^2 + y_1 y_2 \\ \dot{y}_2 &= u_1\end{aligned}\tag{21}$$

(1) 改變內層子系統 Lyapunov 函數 $V_1(y_1)$ 和正定函數 $V_a(y_1)$

設計內層結構的控制律

在設計的第一步驟中，我們在第一小題所選擇之 Lyapunov 函數為一二次函數，在此我們可以選擇更高次方的 Lyapunov 函數，取需為正定，我們的選擇如下

$$V_1'(y_1) = \frac{1}{4} y_1^4\tag{44}$$

將(23)對系統狀態做一次導數，且控制律 $y_2 = \phi'(y_1)$

$$\begin{aligned}\dot{V}_1'(y_1) &= y_1^3 \dot{y}_1 = y_1^3 [y_1^2 + y_1 \phi'(y_1)] \\ &= y_1^5 + y_1^4 \phi'(y_1) \leq -V_a'(y_1) < 0\end{aligned}\tag{45}$$

此時我們可以發現若是正定函數 $V'_a(y_1)$ 還是依照第一小題的選擇 $V'_a(y_1) = y_1^4$ ，得到的內層控制律 $\phi'(y_1) = -1 - y_1$ ，這個情況會使 $\phi(0) = -1 \neq 0$ ，也就是系統無法收斂到原點。若要保持收斂到原點這個特性，就必須跟著改變正定函數 $V_a(y_1)$ 的選擇，觀察式(45)，我們選擇另一種正定函數 $V'_a(y_1) = y_1^6 > 0, \forall y_1 \neq 0$ ，此時內層控制律為

$$\phi'(y_1) = -y_1 - y_1^2 \quad (46)$$

此內層控制律和第一小題所得到的模一樣，但是因為 Lyapunov 函數的改變，最後還是會改變全系統的控制律 $u(y_1, y_2)$ ，根據一般形式的逆向步進控制，全系統的穩定控制律被內層子系統控制律和相應的 Lyapunov 函數，如下

$$u = \frac{1}{g_1(y_1, y_2)} \left[\frac{\partial \phi_1}{\partial y_1} [f_0(y_1) + g_0(y_1)y_2] - \frac{\partial V_1}{\partial y_1} g_0(y_1) - k[y_2 - \phi_1(y_1)] - f_1(y_1, y_2) \right] \quad (47)$$

由此可推導在新的 Lyapunov 函數下，新的控制律 $u'(y_1, y_2)$ 為

$$u'(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^4 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \quad (48)$$

和原先的控制律 $u(y_1, y_2)$ 相比

$$u(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^2 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \quad (38)$$

可以看到將內層子系統的 Lyapunov 函數階數增加至 4 次方，會影響到全系統控制律(47)

中的 $\frac{\partial V_1}{\partial y_1} g_0(y_1)$ 項次，為了多方比較，我們可以再選擇一個新的內層子系統 Lyapunov

函數，並且把階數提高至 6 次方，如下

$$V_1''(y_1) = \frac{1}{6} y_1^6 \quad (49)$$

選擇新正定函數 $V_a''(y_1) = y_1^8 > 0, \forall y_1 \neq 0$ 依照上述的設計方法，可以得到新的全系統控制律 $u''(y_1, y_2)$

$$u''(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^6 - k(y_2 + y_1 + y_1^2) - 3y_1 \right] \quad (50)$$

表 2 整理了三種設計方法和結果。而圖 13、14 則是 MATLAB 模擬結果。此次模擬我們選用兩個初始點，分別是 $(y_1, y_2) = (5, -5)$ 和 $(y_1, y_2) = (-5, 5)$ ，模擬時長設為 5 秒。因時間響應圖能更清楚的看到系統狀態隨時間的變化，故選擇繪製時間響應圖，而不是相平面軌跡圖。圖 13 為狀態變數 y_1 的時間響應圖，共有 6，同一個顏色代表相同的子系統 Lyapunov 函數，一個顏色兩條代表兩個不同的初始點。圖 14 為狀態變數 y_2 的時

間響應圖，共有 6，同一個顏色代表相同的子系統 Lyapunov 函數，一個顏色兩條代表兩個不同的初始點。

表 2、三種內層 Lyapunov 函數、內層正定函數和相應的全系統控制律設計結果

內層 Lyapunov 函數	內層正定函數	全系統控制律
$V_1(x_1) = \frac{1}{2}x_1^2$	$V_a(y_1) = y_1^4$	$u(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^2 \right] - k(y_2 + y_1 + y_1^2) - 3y_1$
$V_1'(y_1) = \frac{1}{4}y_1^4$	$V_a'(y_1) = y_1^6$	$u'(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^4 \right] - k(y_2 + y_1 + y_1^2) - 3y_1$
$V_1''(y_1) = \frac{1}{6}y_1^6$	$V_a''(y_1) = y_1^8$	$u''(y_1, y_2) = \frac{1}{1+(y_2-2)^2} \left[(-1-2y_1)(y_1^2 + y_1y_2) - y_1^6 \right] - k(y_2 + y_1 + y_1^2) - 3y_1$

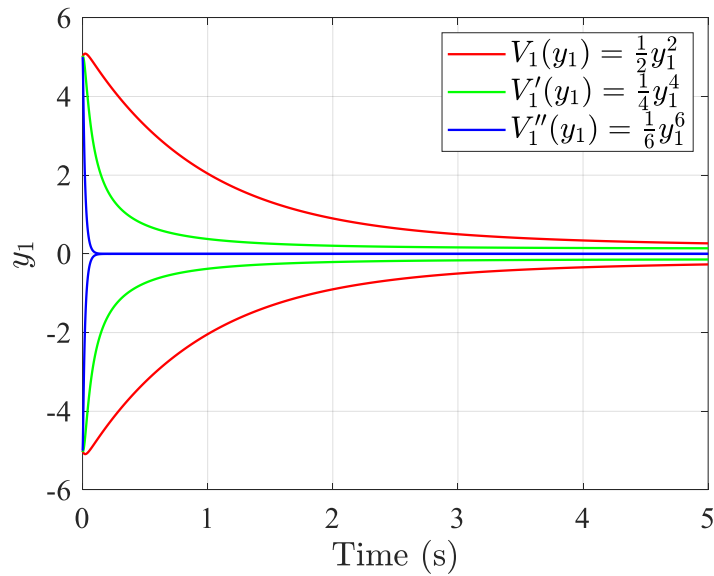


圖 13、不同內層 Lyapunov 函數之系統狀態變數 y_1 時間響應圖

由圖 13 可以看到，提高內層 Lyapunov 函數會增加狀態變數 y_1 的收斂速度，且解決了原先收斂不完全的問題。看似解決了原先的問題，還提高了系統的收斂速度，但是其代價可以在狀態變數 y_2 的時間響應圖中發現。

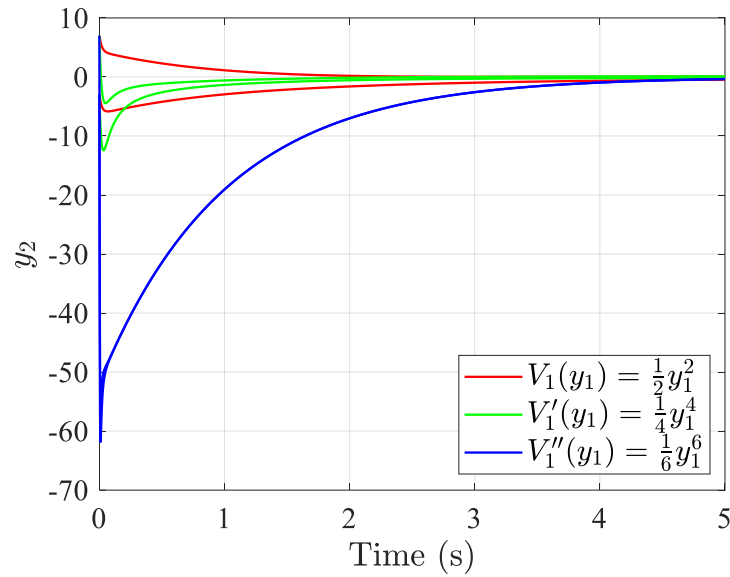


圖 14、不同內層 Lyapunov 函數之系統狀態變數 y_2 時間響應圖

由圖 14 中可以看到，雖然增加內層 Lyapunov 函數的階數提高了狀態變數 y_1 的收斂速度，但是會在狀態變數 y_2 中產生很大的超越量(Overshoot)，這個情況在 $V_1''(y_1)$ 尤其嚴重，其狀態變數 y_1 的收斂速度雖然遠大於其他兩種，但是 y_2 之超越量也遠大於其他兩種。圖 15 為將圖時間拉近至 0 秒~1.5 秒，更可以清楚看到其差距。

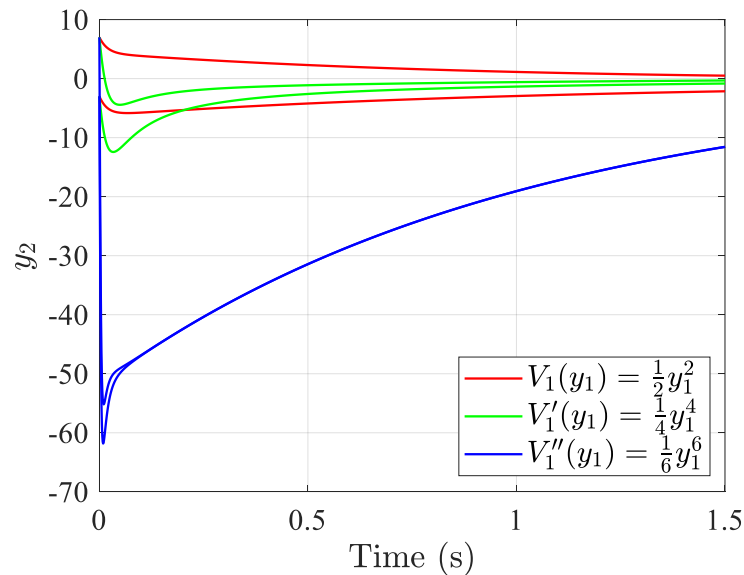


圖 15、不同內層 Lyapunov 函數之系統狀態變數 y_2 時間響應圖 (放大)

此種結果無法用單一標準衡量好壞，而是需要依據實際的需求來選擇哪種內層 Lyapunov 函數較為合適。

(2) 改變控制律 $v = -y_1^2 - kz$ 中的 k 值

改變控制律 $v = -y_1^2 - kz$ 中 k 的選擇也可以改變最後全系統的控制律。為了比較不同 k 值之間的區別，我們採用(48)式中的全系統控制律來做 MATLAB 模擬。此次模擬選擇一個初始點 $(y_1, y_2) = (5, -5)$ 來進行，共有四種 k 值，分別為 1、10、100 和 1000。結果如圖 16~19 所示。因此模擬之暫態響應和穩態可以看到不同的現象，故圖 16、17 繪製 0.5 秒內 (y_1, y_2) 的時間響應，圖 18、19 則是 10 秒內。

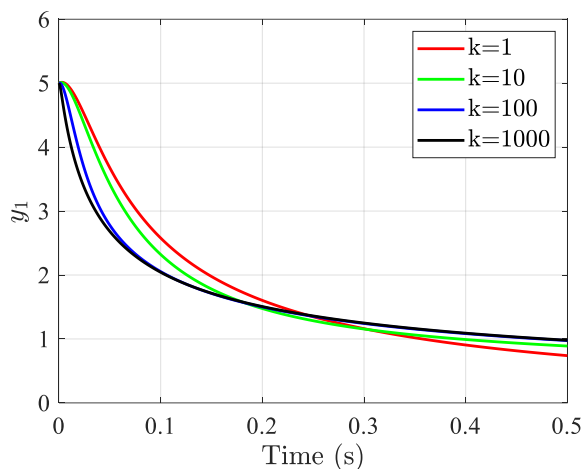


圖 16、不同 k 值， y_1 的時間響應(0.5 秒)

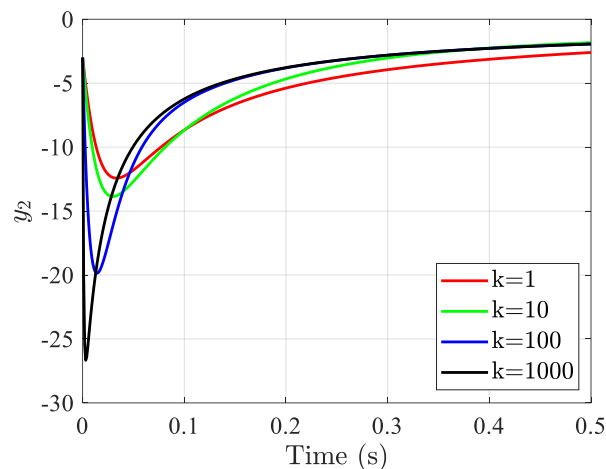


圖 17、不同 k 值， y_2 的時間響應(0.5 秒)

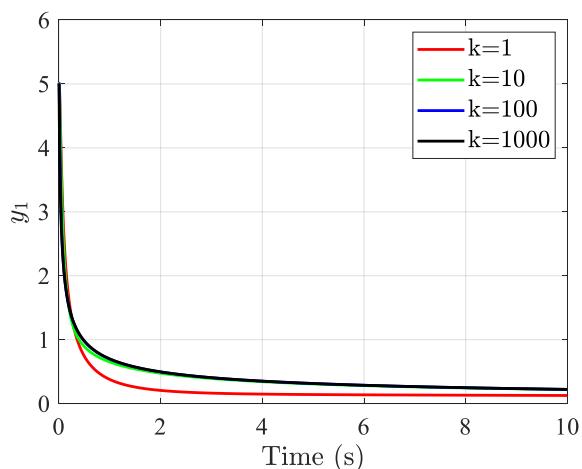


圖 18、不同 k 值， y_1 的時間響應(10 秒)

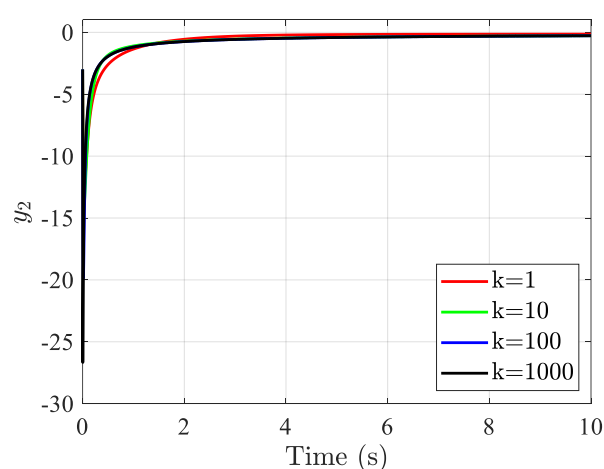


圖 19、不同 k 值， y_2 的時間響應(10 秒)

由圖 16、17 中可以看到，系統從初始點釋放後有一小段時間，從 y_1 看，越大的 k 值收斂越快；從 y_2 看，越大的 k 值有越大的超越量。但從圖 18、19 中可以看到，越小的 k 值在穩態中反而越趨近於平衡點。和前面所述一樣，此種結果無法用單一標準衡量好壞，而是需要依據實際的需求來選擇哪種 k 值較為合適。

MATLAB Code

Question 2

```
%% Nonlinear Control HW7-2
clc;
clear;
close all;

%%
dt=0.001;
t_final=20;
t=0:dt:t_final;
LW=1.2;
FS1=16;
FS_lg=18;
Dfcolor=winter(10);

%% Set Initial Condition
xi_1=[1;1]; xi_2=[0.5;3]; xi_3=[-1;-3]; xi_4=[-2;-4]; xi_5=[2;-3];
xi_6=[0.7;-2]; xi_7=[-4;1]; xi_8=[-0.9;2]; xi_9=[2.3;2.7]; xi_10=[4;-3];
Xi_0=[xi_1, xi_2, xi_3, xi_4, xi_5, xi_6, xi_7, xi_8, xi_9, xi_10];
for i=1:length(Xi_0(1,:))
    Yi_0(:,i)=[Xi_0(1,i);Xi_0(2,i)+2];
end

%% Plot Phase Portrait
f1 = figure ;
for i = 1:length(Xi_0(1,:))
    [ t , Xt1 ] = RK4( @(t,X) Nonlinear_System_Orig1(X) , [0 t_final], Xi_0(:,i) ,dt) ;
    xt1_1_c = Xt1(:,1) ; xt1_2_c = Xt1(:,2) ;
    all_x1(:,i)=xt1_1_c;
    all_x2(:,i)=xt1_2_c;
    plot(xt1_1_c,xt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(xt1_1_c(1,1),xt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-10 10],[0 0], '--', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot([0 0],[-10 15], '--', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot(0,-2, '*', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
ax(1) = gca ;
title('$\mathbf{x}_e = (0,-2)$', 'Interpreter','Latex')
xlabel('$x_1$', 'Interpreter','Latex') % x label
ylabel('$x_2$', 'Interpreter','Latex') % y label
axis([ -6 6 -8 4 ])
xticks(-6:2:6)
yticks(-8:2:4)
axis square
```

```

f2 = figure ;
for i = 1:length(Yi_0(1,:))
    [ t , Yt1 ] = RK4( @(t,Y) Nonlinear_System_CT1(Y) , [0 t_final], Yi_0(:,i) ,dt) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;
    all_y1(:,i)=yt1_1_c;
    all_y2(:,i)=yt1_2_c;
    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-10 10],[0 0], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot([0 0],[-10 15], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot(0,0, '*', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
ax(2) = gca ;
title('$\bar{\mathbf{x}}_e = (0,0)$', 'Interpreter','Latex')
xlabel('$y_1$', 'Interpreter','Latex') % x label
ylabel('$y_2$', 'Interpreter','Latex') % y label
axis([ -6 6 -6 6 ])
xticks(-6:2:6)
yticks(-6:2:6)
axis square

%% Plot Time Response
f3 = figure ;
for i=1:size(all_x1,2)
    plot(t, (all_x1(:,i))', 'Color',Dfcolor(i,:), 'LineWidth',LW);
    hold on
end
xlabel('Time (s)', 'Interpreter','Latex')
ylabel('$x_1$', 'Interpreter','Latex')
xlim([0 t_final])
ylim([-6 6])
ax(3)=gca;
grid on

f4 = figure ;
for i=1:size(all_x2,2)
    plot(t, (all_x2(:,i))', 'Color',Dfcolor(i,:), 'LineWidth',LW);
    hold on
end
xlabel('Time (s)', 'Interpreter','Latex')
ylabel('$x_2$', 'Interpreter','Latex')
xlim([0 t_final])
ylim([-8 5])
ax(4)=gca;
grid on

```

```

f5 = figure ;
for i=1:size(all_y1,2)
    plot(t, (all_y1(:,i)), 'Color',Dfcolor(i,:), 'LineWidth',LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_1$', 'Interpreter', 'Latex')
xlim([0 t_final])
ylim([-6 6])
ax(5)=gca;
grid on

f6 = figure ;
for i=1:size(all_y1,2)
    plot(t, (all_y2(:,i)), 'Color',Dfcolor(i,:), 'LineWidth',LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
xlim([0 t_final])
ylim([-6 6])
ax(6)=gca;
grid on

%%
for i = 1:length(ax)
    set(ax(i), 'XGrid','on', 'YGrid','on', 'FontSize',FS1, 'FontName','Times New Roman')
end
%% Nonlinear System Function
function [dX,Phi,u]=Nonlinear_System_Orig1(X)
x1=X(1);
x2=X(2);
k=2;

y1=x1;
y2=x2+2;
Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^2-k*(y2+y1+y1^2)-3*y1);

dx1=2*x1+x1^2+x1*x2;
dx2=3*x1+(1+x2^2)*u;
dX=[dx1;dx2];
end

function [dY,Phi,u]=Nonlinear_System_CT1(Y)

```

```

y1=Y(1);
y2=Y(2);
k=2;

Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^2-k*(y2+y1+y1^2)-3*y1);

dy1=y1^2+y1*y2;
dy2=3*y1+(1+(y2-2)^2)*u;
dY=[dy1;dy2];
end

```

Question 3

```

%% Nonlinear Control HW7-3

```

```

clc;
clear;
close all;

%%
dt=0.001;
t_final=50;
t=0:dt:t_final;
LW=1.2;
FS1=16;
FS_lg=18;
Dfcolor=winter(17);
nPoint=16;

```

```

%% Set Initial Condition

```

```

for i=0:nPoint
    r=3;
    theta=2*pi/nPoint*i;
    x=r*cos(theta);
    y=r*sin(theta);
    initialPoint_1(1,i+1)=x;
    initialPoint_1(2,i+1)=y;
end

```

```

end
for i=0:nPoint
    r=20;
    theta=2*pi/nPoint*i;
    x=r*cos(theta);
    y=r*sin(theta);
    initialPoint_2(1,i+1)=x;
    initialPoint_2(2,i+1)=y;
end

```

```

end
for i=0:nPoint

```



```

r=150;
theta=2*pi/nPoint*i;
x=r*cos(theta);
y=r*sin(theta);
initialPoint_3(1,i+1)=x;
initialPoint_3(2,i+1)=y;
end
for i=0:nPoint
    r=1000;
    theta=2*pi/nPoint*i;
    x=r*cos(theta);
    y=r*sin(theta);
    initialPoint_4(1,i+1)=x;
    initialPoint_4(2,i+1)=y;
end

for i=0:nPoint
    r=0.2;
    theta=2*pi/nPoint*i;
    x=r*cos(theta);
    y=r*sin(theta);
    initialPoint_5(1,i+1)=x;
    initialPoint_5(2,i+1)=y;
end

%% Plot Phase Portrait
f1 = figure ;
for i = 1:size(initialPoint_1,2)
    [ t , Yt1 ] = RK4( @(t,Y) Nonlinear_System_CT1(Y) , [0 t_final], initialPoint_1(:,i) ,dt) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-10 10],[0 0], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot([0 0],[-10 15], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot(0,0, '*', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
ax(1) = gca ;
xlabel('$y_1$', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
axis([ -6 6 -6 6 ])
xticks(-6:2:6)
yticks(-6:2:6)
axis square

f2 = figure ;

```

```

for i = 1:size(initialPoint_2,2)
    [ t , Yt1 ] = ode45( @(t,Y) Nonlinear_System_CT1(Y) , t , initialPoint_2(:,i)) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-40 40],[0 0], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot([0 0],[-40 40], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot(0,0, '*', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
ax(2) = gca ;
xlabel('$y_1$', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
axis([ -30 30 -30 30 ])
xticks(-30:10:30)
yticks(-30:10:30)
axis square

f3 = figure ;
for i = 1:size(initialPoint_3,2)
    [ t , Yt1 ] = ode45( @(t,Y) Nonlinear_System_CT1(Y) , t , initialPoint_3(:,i)) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-200 200],[0 0], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot([0 0],[-200 200], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
plot(0,0, '*', 'Color',[0.5 0.5 0.5], 'LineWidth',LW) ;
ax(3) = gca ;
xlabel('$y_1$', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
axis([ -200 200 -200 200 ])
xticks(-200:50:200)
yticks(-200:50:200)
axis square

f4 = figure ;
for i = 1:size(initialPoint_4,2)
    [ t , Yt1 ] = ode45( @(t,Y) Nonlinear_System_CT1(Y) , t , initialPoint_4(:,i)) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-1200 1200],[0 0], '--','Color',[0.5 0.5 0.5], 'LineWidth',LW) ;

```

```

plot([0 0],[-1200 1200],'--','Color',[0.5 0.5 0.5],'LineWidth',LW) ;
plot(0,0,'*','Color',[0.5 0.5 0.5],'LineWidth',LW) ;
ax(4) = gca ;
xlabel('$y_1$', 'Interpreter','Latex')
ylabel('$y_2$', 'Interpreter','Latex')
axis([ -1200 1200 -1200 1200 ])
xticks(-1200:400:1200)
yticks(-1200:400:1200)
axis square

f5 = figure ;
for i = 1:size(initialPoint_5,2)
    [ t , Yt1 ] = ode45( @(t,Y) Nonlinear_System_CT1(Y) , t , initialPoint_5(:,i)) ;
    yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

    plot(yt1_1_c,yt1_2_c,'Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
    plot(yt1_1_c(1,1),yt1_2_c(1,1),'o','Color',Dfcolor(i,:), 'LineWidth',LW) ; hold on ;
end
plot([-0.5 0.5],[0 0],'--','Color',[0.5 0.5 0.5],'LineWidth',LW) ;
plot([0 0],[-0.5 0.5],'--','Color',[0.5 0.5 0.5],'LineWidth',LW) ;
plot(0,0,'*','Color',[0.5 0.5 0.5],'LineWidth',LW) ;
ax(5) = gca ;
xlabel('$y_1$', 'Interpreter','Latex')
ylabel('$y_2$', 'Interpreter','Latex')
axis([ -0.3 0.3 -0.3 0.3 ])
xticks(-0.5:0.2:0.5)
yticks(-0.5:0.2:0.5)
axis square

%%
for i = 1:length(ax)
    set(ax(i),'XGrid','on','YGrid','on','FontSize',FS1,'FontName','Times New Roman')
end
%% Nonlinear System Function
function [dY,Phi,u]=Nonlinear_System_CT1(Y)
y1=Y(1);
y2=Y(2);
k=1;

Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^2-k*(y2+y1+y1^2)-3*y1);

dy1=y1^2+y1*y2;
dy2=3*y1+(1+(y2-2)^2)*u;
dY=[dy1;dy2];
end

```

Question 5

%% Nonlinear Control HW7-5

clc;

clear;

close all;

%%

dt=0.001;

t_final=10;

t=0:dt:t_final;

LW=1.8;

FS1=16;

FS_lg=16;

%% Set Initial Condition

xi_1=[5;-5];

xi_2=[-5;5];

Xi_0=[xi_1, xi_2];

for i=1:length(Xi_0(1,:))

 Yi_0(:,i)=[Xi_0(1,i);Xi_0(2,i)+2];

end

K=[1,10,100,1000];

%% For different Lyapunov Function

for i = 1:length(Yi_0(1,:))

 [t , Yt1] = RK4(@(t,Y) Nonlinear_System_CT1(Y,K(1)) , [0 t_final], Yi_0(:,i) ,dt) ;

 yt1_1_c = Yt1(:,1) ; yt1_2_c = Yt1(:,2) ;

 all_y1_1(:,i)=yt1_1_c;

 all_y2_1(:,i)=yt1_2_c;

end

for i = 1:length(Yi_0(1,:))

 [t , Yt2] = RK4(@(t,Y) Nonlinear_System_CT2(Y,K(1)) , [0 t_final], Yi_0(:,i) ,dt) ;

 yt2_1_c = Yt2(:,1) ; yt2_2_c = Yt2(:,2) ;

 all_y1_2(:,i)=yt2_1_c;

 all_y2_2(:,i)=yt2_2_c;

end

for i = 1:length(Yi_0(1,:))

 [t , Yt3] = RK4(@(t,Y) Nonlinear_System_CT3(Y,K(1)) , [0 t_final], Yi_0(:,i) ,dt) ;

 yt3_1_c = Yt3(:,1) ; yt3_2_c = Yt3(:,2) ;

 all_y1_3(:,i)=yt3_1_c;

 all_y2_3(:,i)=yt3_2_c;

end

%% Plot Time Response

f1=figure;

```

for i=1:size(all_y1_1,2)
    p(1)=plot(t, (all_y1_1(:,i))', 'r', 'LineWidth', LW);
    hold on
end
for i=1:size(all_y1_2,2)
    p(2)=plot(t, (all_y1_2(:,i))', 'g', 'LineWidth', LW);
    hold on
end
for i=1:size(all_y1_3,2)
    p(3)=plot(t, (all_y1_3(:,i))', 'b', 'LineWidth', LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_1$', 'Interpreter', 'Latex')
xlim([0 t_final])
ylim([-6 6])
hs(1)=legend([p(1),p(2),p(3)], {'$V_1(y_1)=\frac{1}{2}y_1^2$',
'$V_1^{\prime}(y_1)=\frac{1}{4}y_1^4$', '$V_1^{\prime\prime}(y_1)=\frac{1}{6}y_1^6$'} ,
'Interpreter', 'latex');
ax(1)=gca;
grid on

f2=figure;
for i=1:size(all_y2_1,2)
    p(1)=plot(t, (all_y2_1(:,i))', 'r', 'LineWidth', LW);
    hold on
end
for i=1:size(all_y2_2,2)
    p(2)=plot(t, (all_y2_2(:,i))', 'g', 'LineWidth', LW);
    hold on
end
for i=1:size(all_y2_3,2)
    p(3)=plot(t, (all_y2_3(:,i))', 'b', 'LineWidth', LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
xlim([0 5])
ylim([-70 10])
hs(2)=legend([p(1),p(2),p(3)], {'$V_1(y_1)=\frac{1}{2}y_1^2$',
'$V_1^{\prime}(y_1)=\frac{1}{4}y_1^4$', '$V_1^{\prime\prime}(y_1)=\frac{1}{6}y_1^6$'} , 'In
terpreter', 'latex', 'location', 'Southeast');
ax(2)=gca;
grid on

%% For different k value

```

```

f3=figure;
for i = 1:length(K)
%   [ t , Yt4 ] = RK4( @(t,Y) Nonlinear_System_CT2(Y,K(i)) , [0 t_final], Yi_0(:,1) ,dt) ;
    [ t , Yt4 ] = ode45( @(t,Y) Nonlinear_System_CT2(Y,K(i)) , t , Yi_0(:,1)) ;
    yt4_1_c = Yt4(:,1) ; yt4_2_c = Yt4(:,2) ;
    all_y1_4(:,i)=yt4_1_c;
    all_y2_4(:,i)=yt4_2_c;
end
Dfcolor=[1 0 0;0 1 0;0 0 1;0 0 0];
for i=1:length(K)
    p(i+3)=plot(t, (all_y1_4(:,i))' , 'Color',Dfcolor(i,:), 'LineWidth', LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_1$', 'Interpreter', 'Latex')
xlim([0 t_final])
hs(3)=legend([p(4),p(5),p(6),p(7)], {'k=1' , 'k=10' , 'k=100' , 'k=1000'} , 'Interpreter', 'latex');
ax(3)=gca;
grid on

f4=figure;
for i=1:length(K)
    p(i+7)=plot(t, (all_y2_4(:,i))' , 'Color',Dfcolor(i,:), 'LineWidth', LW);
    hold on
end
xlabel('Time (s)', 'Interpreter', 'Latex')
ylabel('$y_2$', 'Interpreter', 'Latex')
xlim([0 t_final])
ylim([-30 1])
hs(4)=legend([p(8),p(9),p(10),p(11)], {'k=1' , 'k=10' , 'k=100' , 'k=1000'} , 'Interpreter', 'latex',
'location', 'Southeast');
ax(4)=gca;
grid on
%%
for i = 1:length(ax)
    set(ax(i), 'XGrid', 'on', 'YGrid', 'on', 'FontSize', FS1, 'FontName', 'Times New Roman')
end
for i = 1:length(hs)
    set(hs(i), 'FontSize', FS_lg, 'FontName', 'Times New Roman')
end
%% Nonlinear System Function
function [dY,Phi,u]=Nonlinear_System_CT1(Y,K)
y1=Y(1);
y2=Y(2);
k=K;

```

```

Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^2-k*(y2+y1+y1^2)-3*y1);

dy1=y1^2+y1*y2;
dy2=3*y1+(1+(y2-2)^2)*u;
dY=[dy1;dy2];
end

function [dY,Phi,u]=Nonlinear_System_CT2(Y,K)
y1=Y(1);
y2=Y(2);
k=K;

Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^4-k*(y2+y1+y1^2)-3*y1);

dy1=y1^2+y1*y2;
dy2=3*y1+(1+(y2-2)^2)*u;
dY=[dy1;dy2];
end

function [dY,Phi,u]=Nonlinear_System_CT3(Y,K)
y1=Y(1);
y2=Y(2);
k=K;

Phi=-y1-y1^2;
u=(1/(1+(y2-2)^2))*((-1-2*y1)*(y1^2+y1*y2)-y1^6-k*(y2+y1+y1^2)-3*y1);

dy1=y1^2+y1*y2;
dy2=3*y1+(1+(y2-2)^2)*u;
dY=[dy1;dy2];
end

```