



6 Método de Steepest Descent

- Se discutirá un esquema de adaptación con base en información de gradiente a través de la descripción de un método de optimización clásico conocido como **método de gradientes conjugados**.
- Este método es básico para el entendimiento de varias formas prácticas de implementar algoritmos de adaptivos con base en el gradiente.
- El método de gradiente es recursivo en si mismo, ya que comenzando en algún valor inicial arbitrario del vector de parámetros evoluciona con el incremento del número de iteraciones.
- El valor final así calculado converge a la solución de Wiener.
- El aspecto importante a notar con este método es que describe un **sistema determinístico realimentado** que encuentra el mínimo de la superficie de desempeño sin conocimiento específico de la misma. De esta forma, provee ideas heurísticas para el algoritmo least mean squares (LMS), que será discutido posteriormente.

6.1 Algunos conceptos

- Consideraremos un filtro transversal con entrada $u(n)$ (proceso ESA de media cero y matriz correlación \mathbf{R}) y el correspondiente conjunto $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T$, además de $d(n)$ lo que provee un marco de referencia para la acción de filtrado óptimo.

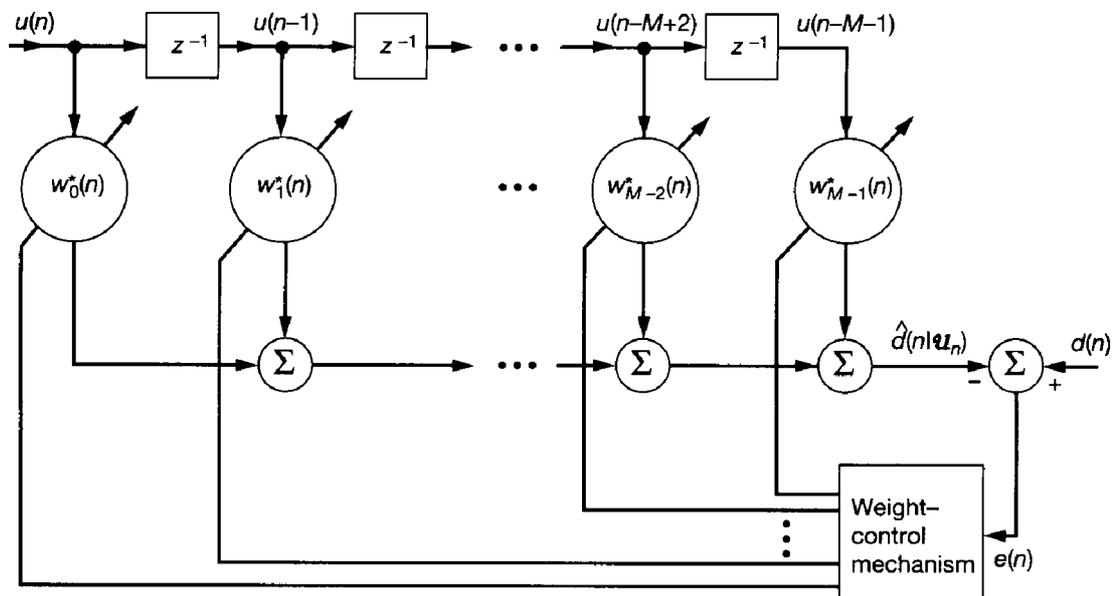


Figura 17: Estructura del filtro adaptivo transversal

- Con vector de entrada $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T$ y la estimación de la respuesta deseada como $\hat{d}(n|\mathcal{U}_n)$.
- El error de estimación se definirá como

$$e(n) = d(n) - \hat{d}(n|\mathcal{U}_n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n)$$

- Si $\mathbf{u}(n)$ y $d(n)$ son conjuntamente ESA, entonces el **error medio cuadrático** ó la **función costo** $J(n)$ en el instante n es una función cuadrática del vector de coeficientes, como mostrado por

$$J(n) = \sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n) \quad (57)$$

- La ecuación (57) define el error medio cuadrático que resultaría si el vector de coeficientes del filtro fuera fijo en el valor $\mathbf{w}(n)$. Dado que $\mathbf{w}(n)$ varía con n , es natural que el error medio cuadrático varía con n en la forma correspondiente. Esta variación de $J(n)$ significa que el proceso de error de estimación $e(n)$ no es estacionario.
- Es posible visualizar la relación entre los coeficientes del filtro y $J(n)$ como una **superficie paraboloidal** con un único mínimo, que denominaremos **superficie de desempeño del error** asociado al filtro adaptivo. El proceso adaptivo tiene como propósito buscar el punto de mínimo de esta superficie.
- En el punto de mínimo de la superficie de desempeño el vector de coeficientes tomará el valor óptimo \mathbf{w}_o , definido por las ecuaciones de Wiener-Hopf: $\mathbf{R}\mathbf{w}_o = \mathbf{p}$. El valor del error medio cuadrático mínimo será

$$J_{min} = \sigma_d^2 - \mathbf{p}^H\mathbf{w}_o$$



6.2 Algoritmo Steepest Descent

- El requerimiento para el filtro adaptivo es resolver la ecuación de Wiener-Hopf en términos de los coeficientes asociados. Una forma es resolver el sistema de ecuaciones por medios analíticos directamente.
- Sin embargo, esa forma introduce problemas computacionales serios sobre todo cuando el filtro tiene un gran número de coeficientes y cuando la velocidad de procesamiento requerida es alta.
- Un procedimiento alternativo es a través del **método Steepest Descent** (SD), uno de los antiguos métodos de optimización.
- Para hallar el valor del mínimo del error medio cuadrático J_{min} mediante el algoritmo SD es posible proceder como a continuación
 1. Comenzamos con un valor inicial $\mathbf{w}(0)$ para el vector de coeficientes, el cual provee una estimación inicial del punto de mínimo de la superficie de error. A menos que se disponga de alguna información a priori, usualmente se hace $\mathbf{w}(0) = \mathbf{0}$.
 2. Con la estimación disponible se calcula el **vector gradiente**, cuyas partes real e imaginaria están definidas como la derivada del error medio cuadrático $J(n)$, evaluada con respecto a las partes real e imaginaria del vector de coeficientes $\mathbf{w}(n)$.
 3. Se calcula la próxima estimación del vector de coeficientes haciendo un cambio en la estimación anterior en la dirección opuesta a la del vector gradiente.
 4. Se vuelve al paso 2 y se repite el proceso.

- Si $\nabla J(n)$ es el valor del **vector gradiente** en el instante n entonces el método SD puede escribirse recursivamente como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)] \quad (58)$$

donde μ es una constante positiva real y el factor $1/2$ se incluye para simplificar constantes posteriores en el análisis del método.

- El vector gradiente está dado por

$$\begin{aligned} \nabla J(n) &= \begin{bmatrix} \frac{\partial J(n)}{\partial a_0(n)} + j \frac{\partial J(n)}{\partial b_0(n)} \\ \frac{\partial J(n)}{\partial a_1(n)} + j \frac{\partial J(n)}{\partial b_1(n)} \\ \vdots \\ \frac{\partial J(n)}{\partial a_{M-1}(n)} + j \frac{\partial J(n)}{\partial b_{M-1}(n)} \end{bmatrix} \\ &= -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \end{aligned}$$

donde se supone que \mathbf{R} y \mathbf{p} son conocidos. Sustituyendo el gradiente en (58) se obtiene

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \quad (59)$$

- Es posible observar que el parámetro μ controla el tamaño del paso de la corrección incremental aplicada al vector de coeficientes, se lo denomina en general **factor de convergencia**.

6.3 Estabilidad del algoritmo

- Como el algoritmo es recursivo existe la posibilidad de volverse **inestable**. El **desempeño de estabilidad** del algoritmo SD está determinado por dos factores: μ y \mathbf{R} .
- Para determinar la **condición de estabilidad** del algoritmo es necesario examinar los **modos naturales** del mismo. En particular, utilizaremos la representación de \mathbf{R} en términos de autovalores y autovectores para definir una versión transformada del vector de coeficientes.
- Comenzaremos el análisis definiendo el **vector de error en los coeficientes** como

$$\mathbf{c}(n) = \mathbf{w}(n) - \mathbf{w}_o$$

donde \mathbf{w}_o es el valor óptimo del vector de coeficientes, definido por la ecuación de Wiener-Hopf.

- Luego eliminando el vector de correlación cruzada \mathbf{p} se obtiene

$$\mathbf{c}(n + 1) = (\mathbf{I} - \mu \mathbf{R}) \mathbf{c}(n)$$

- Usando la transformación de similaridad unitaria es posible expresar la matriz correlación como $\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H$.
- De esta forma

$$\mathbf{c}(n + 1) = (\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H) \mathbf{c}(n) \quad (60)$$

- Premultiplicando ambos lados de esta ecuación por \mathbf{Q}^H y usando que $\mathbf{Q}^H = \mathbf{Q}^{-1}$, se tiene que

$$\mathbf{Q}^H \mathbf{c}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{Q}^H \mathbf{c}(n)$$

- Si definimos ahora un nuevo conjunto de coordenadas como

$$\boldsymbol{\nu}(n) = \mathbf{Q}^H \mathbf{c}(n) = \mathbf{Q}^H [\mathbf{w}(n) - \mathbf{w}_o] \quad (61)$$

tal que (60) se transforma en

$$\boldsymbol{\nu}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \boldsymbol{\nu}(n) \quad (62)$$

- El valor inicial de $\boldsymbol{\nu}(n)$ será

$$\begin{aligned} \boldsymbol{\nu}(0) &= \mathbf{Q}^H [\mathbf{w}(0) - \mathbf{w}_o] \\ &= -\mathbf{Q}^H \mathbf{w}_o \end{aligned} \quad (63)$$

donde en el segundo paso se consideró que $\mathbf{w}(0) = \mathbf{0}$. El k -ésimo **modo natural** del algoritmo SD se tiene que

$$\nu_k(n+1) = (1 - \mu \lambda_k) \nu_k(n), \quad k = 1, 2, \dots, M \quad (64)$$

- La ecuación (64) es una **ecuación a diferencias de primer orden**. Suponiendo que $\nu_k(n)$ tiene valor inicial $\nu_k(0)$, la solución tiene la forma

$$\nu_k(n) = (1 - \mu \lambda_k)^n \nu_k(0), \quad k = 1, 2, \dots, M \quad (65)$$

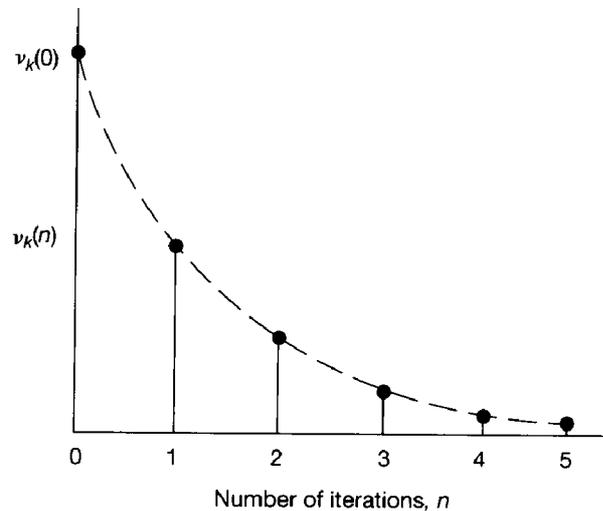


Figura 18: Variación en el tiempo del modo natural k -ésimo en el algoritmo SD con el tiempo, suponiendo que $1 - \mu\lambda_k < 1$.

- Dado que todos los autovalores de \mathbf{R} son positivos y reales, la respuesta $\nu_k(n)$ no exhibe oscilaciones. Además, (ver figura 18), los números generados por (65) representan una **serie geométrica** con razón igual a $1 - \mu\lambda_k$.
- Para garantizar la **estabilidad** o **convergencia** del algoritmo SD, la magnitud de esta razón debe ser menor que 1 para todo k . Esto se verifica si $-1 < 1 - \mu\lambda_k < 1$ (entonces con $n \rightarrow \infty$ todos los modos naturales del algoritmo se desvanecen, independientemente de las condiciones iniciales).
- Esto es equivalente a decir que $\mathbf{w}(n) \rightarrow \mathbf{w}_o$ cuando n tiende a infinito.
- La condición necesaria y suficiente para la convergencia o estabilidad del algoritmo SD es que el factor de convergencia μ satisfaga

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (66)$$

- En base a la figura 18 es posible inferir una envolvente exponencial de **constante de tiempo** τ_k asociada a la serie geométrica. Asumiendo que la unidad de tiempo es la duración de un ciclo de iteración y eligiendo la constante de tiempo τ_k tal que

$$1 - \mu \lambda_k = e^{-\frac{1}{\tau_k}}$$

se obtiene

$$\tau_k = \frac{-1}{\ln(1 - \mu \lambda_k)} \quad (67)$$

- La constante de tiempo τ_k define el número de iteraciones requeridas por la amplitud del k -ésimo modo natural $\nu_k(n)$ para decaer $1/e$ de su valor inicial $\nu_k(0)$. Para el caso especial de adaptación lenta (μ pequeño es posible aproximar la constante de tiempo por

$$\tau_k \cong \frac{1}{\mu \lambda_k}, \quad \mu \ll 1$$

- Es posible formular el comportamiento transitorio del vector de coeficientes premultiplicando ambos lados de (61) por \mathbf{Q} , y resolviendo para $\mathbf{w}(n)$ como sigue

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{w}_o + \mathbf{Q} \boldsymbol{\nu}(n) \\ &= \mathbf{w}_o + \sum_{k=1}^M \mathbf{q}_k \nu_k(n) \end{aligned} \quad (68)$$

- Luego, el comportamiento transitorio del i -ésimo coeficiente como

$$w_i(n) = w_{oi} + \sum_{k=1}^N q_{ki} \nu_k(0) (1 - \mu \lambda_k)^n, \quad i = 1, 2, \dots, M \quad (69)$$

donde w_{oi} es el valor óptimo del i -ésimo coeficiente y q_{ki} es el i -ésimo elemento del k -ésimo autovector \mathbf{q}_k .

- Esta ecuación muestra que cada coeficiente del algoritmo SD converge a una sumatoria ponderada de exponenciales de la forma $(1 - \mu \lambda_k)^n$. El tiempo τ_k requerido por cada término para alcanzar $1/e$ de su valor inicial está dado por (67).
- La **constante de tiempo total**, τ_a , definida como el tiempo requerido por el término de sumatoria de (69) para decaer $1/e$ de su valor inicial, no puede expresarse en forma cerrada.
- Independientemente de eso, la **menor velocidad de convergencia** se logra cuando $q_{ki} \nu_k(0)$ es cero para todo k excepto para el modo correspondiente al menor autovalor λ_{min} de \mathbf{R} , tal que el límite superior de τ_a está definido por $-1/\ln(1 - \mu \lambda_{min})$.
- La **mayor velocidad de convergencia** se logra cuando todos los $q_{ki} \nu_k(0)$ son cero excepto para el modo correspondiente al mayor autovalor λ_{max} , tal que la cotainferior sobre τ_a está definida para $-1/\ln(1 - \mu \lambda_{max})$. En consecuencia, la constante de tiempo total para cualquier coeficiente del algoritmo SD está acotada como sigue:

$$\frac{-1}{\ln(1 - \mu \lambda_{max})} \leq \tau_a \leq \frac{-1}{\ln(1 - \mu \lambda_{min})}$$

Comportamiento del MSE en el transitorio

- Es posible obtener mayores detalles de la operación del algoritmo SD examinando el comportamiento transitorio del error medio cuadrático $J(n)$. Ya vimos que

$$J(n) = J_{min} + \sum_{k=1}^M \lambda_k |\nu_k(n)|^2 \quad (70)$$

donde J_{min} es el error medio cuadrático mínimo.

- El comportamiento transitorio del k -ésimo modo natural, $\nu_k(n)$ está definido por (65), de forma que sustituida en (70) se tiene que

$$J(n) = J_{min} + \sum_{k=1}^M \lambda_k (1 - \mu \lambda_k)^{2n} |\nu_k(0)|^2 \quad (71)$$

- Cuando el algoritmo SD converge (o sea, μ satisface las cotas de (66)), es posible observar que, independientemente de las condiciones iniciales

$$\lim_{n \rightarrow \infty} J(n) = J_{min}$$

- La curva obtenida graficando el error medio cuadrático $J(n)$ versus el número de iteraciones n se denomina **curva de aprendizaje**.
- De (71) vemos que **la curva de aprendizaje del algoritmo SD consiste en la suma de exponenciales, cada una de las cuales corresponde a un modo natural del algoritmo**.

- En general el número de modos iguala al número de coeficientes. Para llegar del valor inicial $J(0)$ al valor final J_{min} , la disminución exponencial del k -ésimo modo tiene una constante de tiempo igual a

$$\tau_{\kappa, mse} \cong \frac{-1}{2 \ln(1 - \mu \lambda_k)}$$

que para pequeños valores de μ puede aproximarse por

$$\tau_{\kappa, mse} \cong \frac{1}{2\mu \lambda_k}$$

y que muestra que cuanto menor es el factor de convergencia μ , menor será la velocidad de disminución de cada modo natural del algoritmo SD.

6.4 Ejemplo

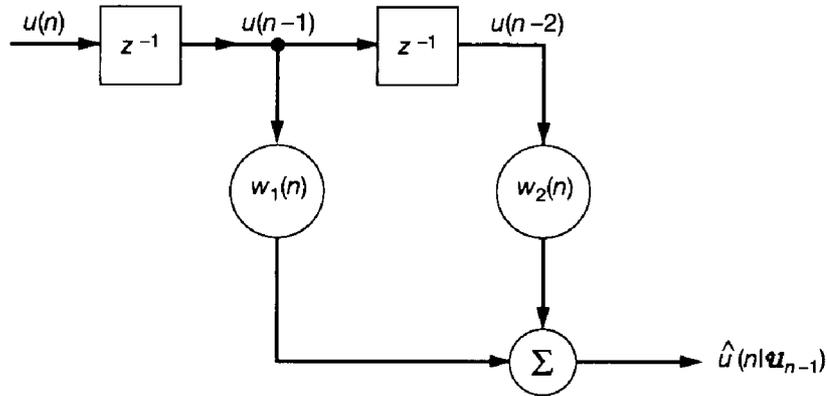


Figura 19: Predictor de segundo orden para una entrada real.

- Se estudiará en este ejemplo el comportamiento transitorio del algoritmo SD aplicado a un predictor que opera sobre un proceso autorregresivo (AR) real. La figura muestra la estructura del predictor, formado por dos coeficientes $w_1(n)$ y $w_2(n)$; la dependencia de esos coeficientes del número de iteraciones n enfatiza las condiciones del transitorio del predictor.
- El proceso AR $u(n)$ se describe por

$$u(n) + a_1 u(n-1) + a_2 u(n-2) = \nu(n)$$

donde $\nu(n)$ es un proceso de ruido blanco de media cero y varianza σ_ν^2 . Los parámetros AR se eligen tal que las raíces de la ecuación característica

$$1 + a_1 z^{-1} + a_2 z^{-2} = 0$$

sean complejas, o sea $a_1^2 < 4a_2$. Los valores particulares de a_1 y a_2 están determinados por la dispersión de los autovalores $\chi(\mathbf{R})$ deseada.

- Para valores de a_1 y a_2 especificados, la varianza σ_v^2 del proceso de ruido blanco se elige de forma que $u(n)$ tenga varianza $\sigma_u^2 = 1$.
- La idea es evaluar el comportamiento transitorio del algoritmo SD para las siguientes condiciones:
 - Variación de la dispersión de los autovalores $\chi(\mathbf{R})$ manteniendo fijo el factor de convergencia μ .
 - Variación del factor de convergencia μ , manteniendo fija la dispersión de autovalores $\chi(\mathbf{R})$.

Caracterización del proceso AR

- Como el predictor tiene dos coeficientes y el proceso AR es real, la matriz correlación \mathbf{R} es de 2×2 simétrica

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix}$$

donde $r(0) = \sigma_u^2$, $r(1) = -\frac{a_1}{1+a_2}\sigma_u^2$, $\sigma_u^2 = \left(\frac{1+a_2}{1-a_2}\right) \frac{\sigma_v^2}{(1+a_2)^2 - a_1^2}$.

- Las autovalores de \mathbf{R} serán

$$\lambda_1 = \left(1 - \frac{a_1}{1+a_2}\right) \sigma_u^2 \quad \lambda_2 = \left(1 + \frac{a_1}{1+a_2}\right) \sigma_u^2$$

de donde, suponiendo a_1 negativo, la dispersión de autovalores es

$$\chi(\mathbf{R}) = \frac{\lambda_1}{\lambda_2} = \frac{1 - a_1 + a_2}{1 + a_1 + a_2}$$

Caso	Parámetros AR		Autovalores		Dispersión	MSE mínimo
	a_1	a_2	λ_1	λ_2	$\chi = \lambda_1/\lambda_2$	$J_{min} = \sigma_\nu^2$
1	-0.1950	0.95	1.1	0.9	1.22	0.0965
2	-0.9750	0.95	1.5	0.5	3	0.0731
3	-1.5955	0.95	1.818	0.182	10	0.0322
4	-1.9114	0.95	1.957	0.0198	100	0.0038

Tabla 1: Parámetros del modelo AR de segundo orden

- Los autovalores \mathbf{q}_1 y \mathbf{q}_2 asociados con λ_1 y λ_2 son

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Experimento 1: Variación de la dispersión de los autovalores. En este experimento se fijó $\mu = 0.3$, y se realizaron evaluaciones para los cuatro conjuntos de parámetros AR de la tabla 1.

Para cada conjunto de parámetros se obtiene un gráfico bidimensional de los coeficientes transformados $\nu_1(n)$ y $\nu_2(n)$ para mostrar el comportamiento transitorio del algoritmo SD. En particular, a partir de (65), se obtiene

$$\boldsymbol{\nu}(n) = \begin{bmatrix} \nu_1(n) \\ \nu_2(n) \end{bmatrix} = \begin{bmatrix} (1 - \mu\lambda_1)^n \nu_1(0) \\ (1 - \mu\lambda_2)^n \nu_2(0) \end{bmatrix}, \quad n = 1, 2, \dots \quad (72)$$

- Para calcular el valor inicial $\nu(0)$ se utiliza (63), suponiendo el valor inicial de $\mathbf{w}(0) = \mathbf{0}$, que requiere el conocimiento de \mathbf{w}_o . Entonces, con el proceso de segundo orden AR como entrada del predictor se tiene que

$$\mathbf{w}_o = \begin{bmatrix} -a_1 \\ -a_2 \end{bmatrix}$$

y $J_{min} = \sigma_\nu^2$.

- Siguiendo (63) se tiene que

$$\boldsymbol{\nu}(0) = \begin{bmatrix} \nu_1(0) \\ \nu_2(0) \end{bmatrix} = \frac{-1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -a_1 \\ -a_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} a_1 + a_2 \\ a_1 - a_2 \end{bmatrix} \quad (73)$$

- De esta forma, para los parámetros especificados es posible utilizar (73) para calcular el valor inicial $\boldsymbol{\nu}(0)$ y luego (72) para calcular $\boldsymbol{\nu}(1)$, $\boldsymbol{\nu}(2)$, \dots . Asociando estos puntos para distintos n se obtiene una **trayectoria** que describe el comportamiento transitorio del algoritmo SD.
- Resulta muy informativo incluir en los gráficos bidimensionales el lugar de $\nu_1(n)$ versus $\nu_2(n)$ representado por (70) para valores fijos de n . Para nuestro ejemplo se tiene que

$$J(n) - J_{min} = \lambda_1 \nu_1^2(n) + \lambda_2 \nu_2^2(n) \quad (74)$$

- Cuando $\lambda_1 = \lambda_2$ y n es fijo, esta ecuación representa un círculo con centro en el origen y radio igual a la raíz cuadrada de $[J(n) - J_{min}]/\lambda$, donde λ es el valor común de los dos autovalores. Cuando $\lambda_1 \neq \lambda_2$, la ecuación anterior representa una elipse con eje mayor igual a la raíz cuadrada de $[J(n) - J_{min}]/\lambda_2$ y el eje menor igual a la raíz cuadrada de $[J(n) - J_{min}]/\lambda_1$.

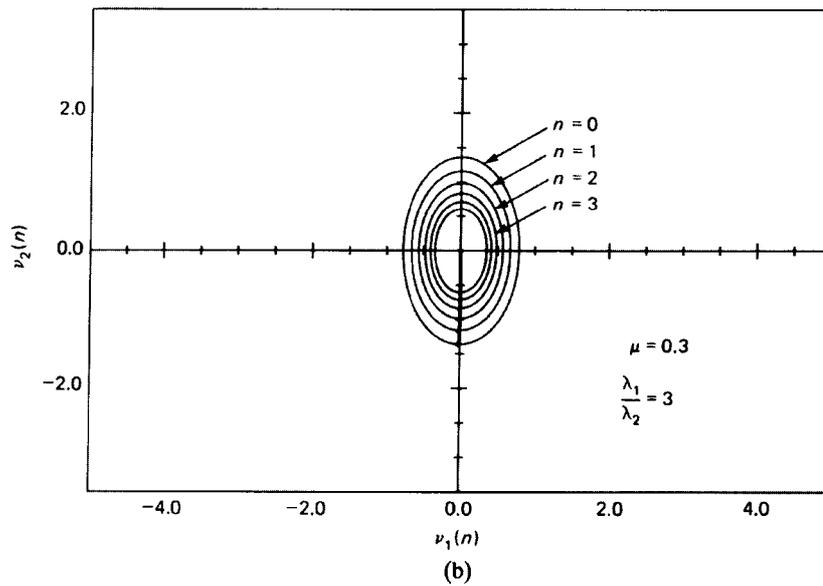
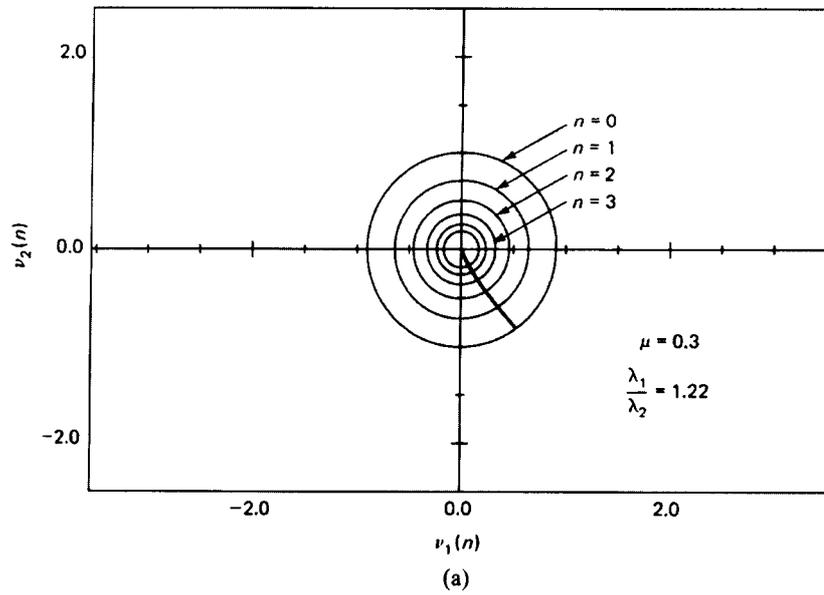


Figura 20: Lugar geométrico de $v_1(n)$ versus $v_2(n)$ para el algoritmo SD con factor de convergencia $\mu = 0.3$ y dispersión de autovalores variable: a) $\chi(\mathbf{R}) = 1.22$.

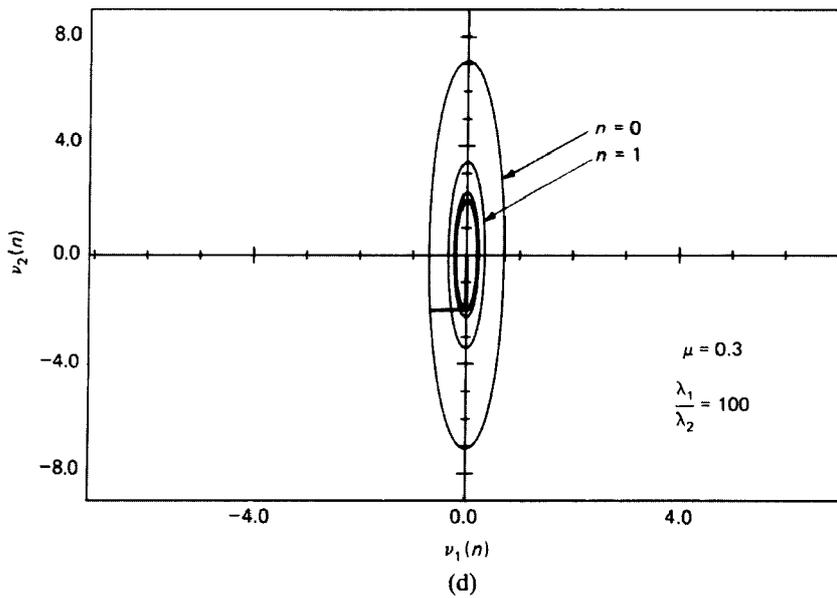
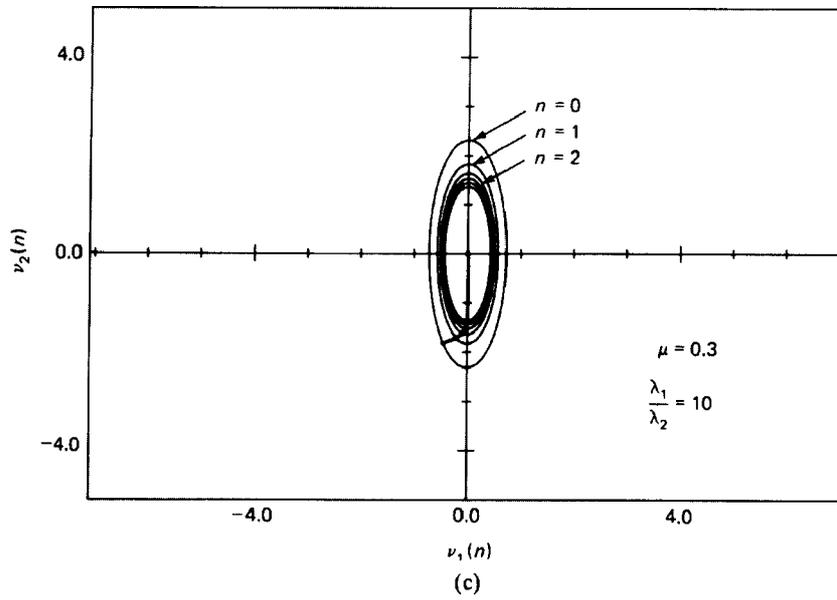


Figura 21: Lugar geométrico de $v_1(n)$ versus $v_2(n)$ para el algoritmo SD con factor de convergencia $\mu = 0.3$ y dispersión de autovalores variable: c) $\chi(\mathbf{R}) = 10$, d) $\chi(\mathbf{R}) = 100$.

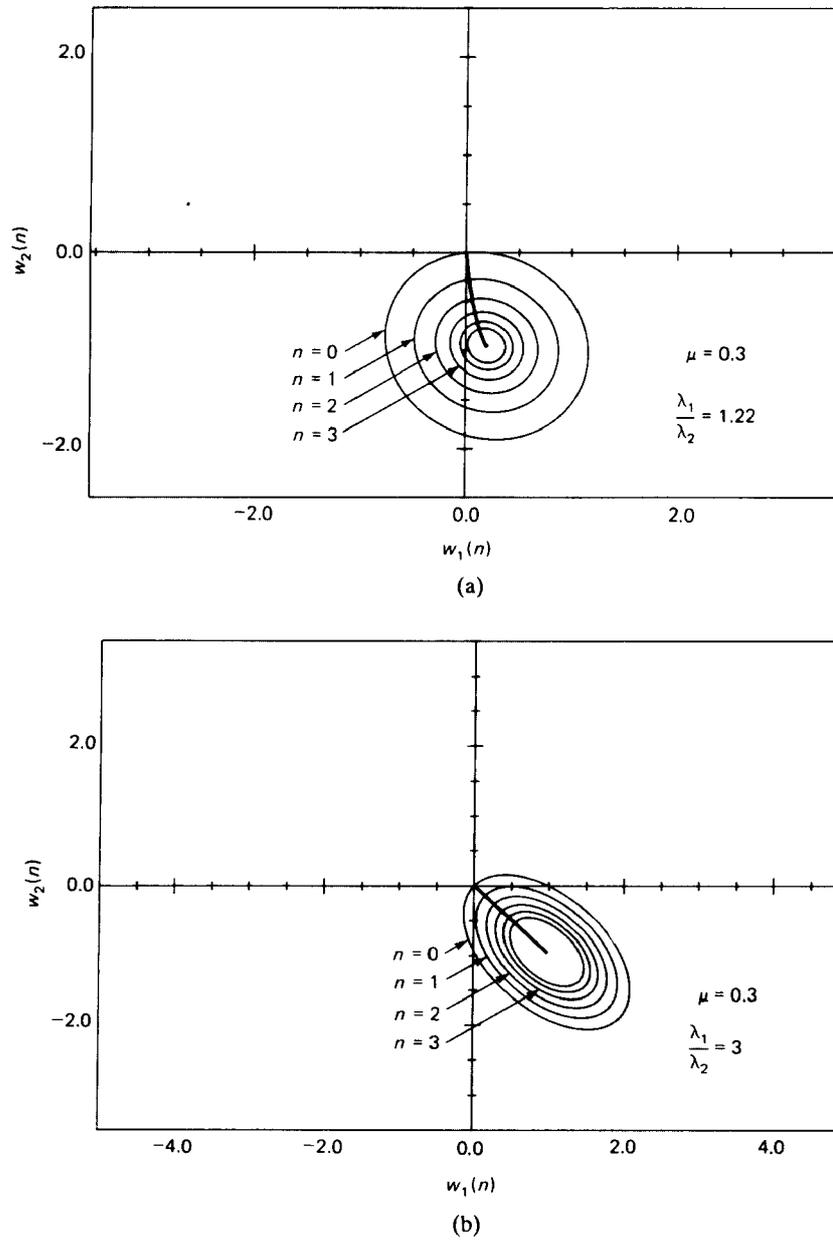
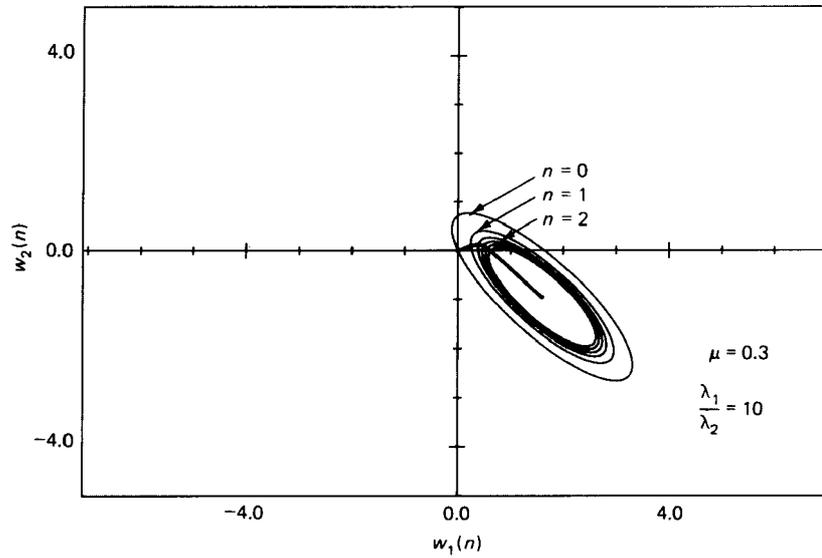
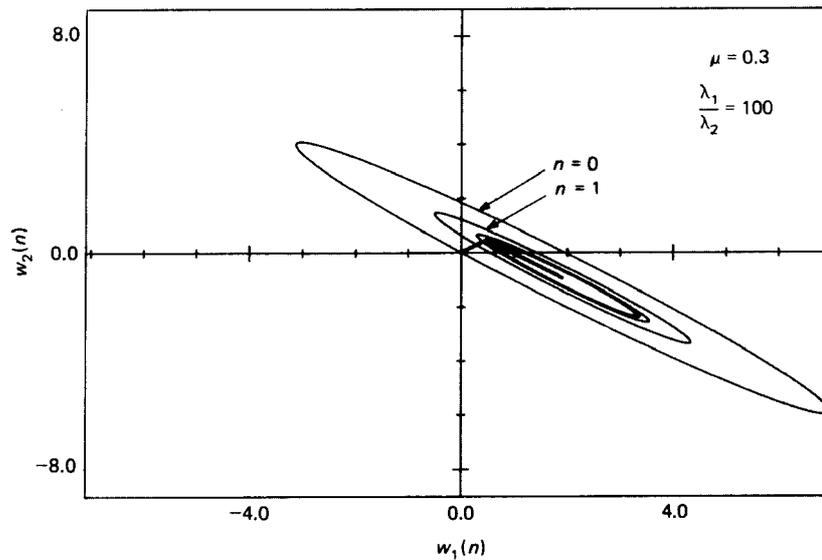


Figura 22: Lugar geométrico de $w_1(n)$ versus $w_2(n)$ para el algoritmo SD con factor de convergencia $\mu = 0.3$ y dispersión de autovalores variable: a) $\chi(\mathbf{R}) = 1.22$, b) $\chi(\mathbf{R}) = 3$.



(c)



(d)

Figura 23: Lugar geométrico de $w_1(n)$ versus $w_2(n)$ para el algoritmo SD con factor de convergencia $\mu = 0.3$ y dispersión de autovalores variable: c) $\chi(\mathbf{R}) = 10$, d) $\chi(\mathbf{R}) = 100$.

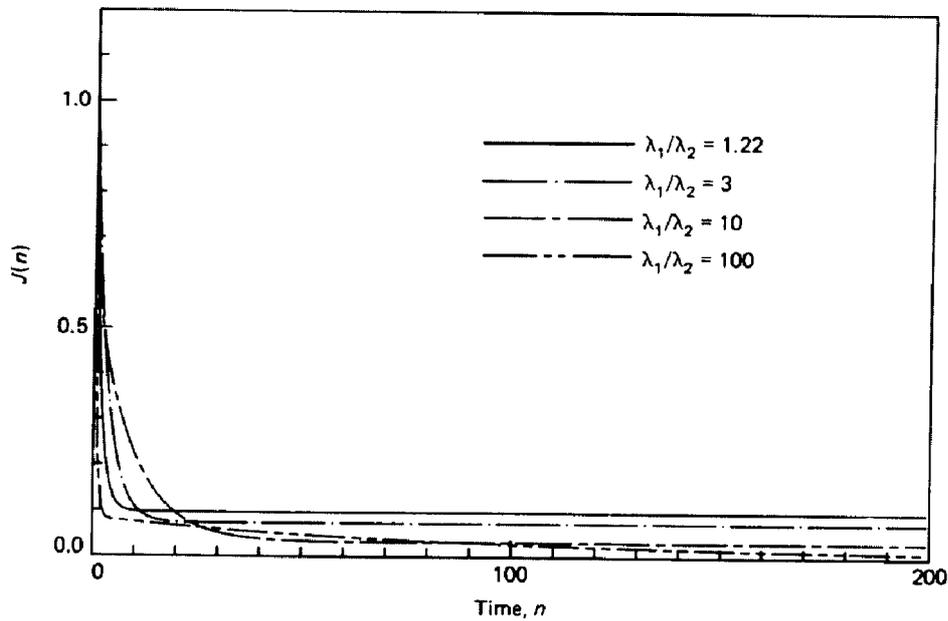


Figura 24: Curvas de aprendizaje para el algoritmo SD con factor de convergencia $\mu = 0.3$ y dispersión de autovalores variable: a) $\chi(\mathbf{R}) = 1.22$, b) $\chi(\mathbf{R}) = 3$, c) $\chi(\mathbf{R}) = 10$, d) $\chi(\mathbf{R}) = 100$.

En la figura 24 se ha graficado $J(n)$ versus n para las diferentes dispersiones de autovalores 1.22, 3, 10 y 100 (cuando la dispersión de autovalores aumenta, y la entrada se vuelve mal correlacionada, el error medio cuadrático mínimo J_{min} disminuye).

Experimento 2: Variación del factor de convergencia. En este experimento se fija $\chi(\mathbf{R}) = 10$ y se varía el factor de convergencia μ . En particular, examinaremos el comportamiento transitorio del algoritmo SD para $mu = 0.3$ y $mu = 1.0$. Los resultados correspondientes en términos de los coeficientes $\nu_1(n)$ y $\nu_2(n)$ se muestran en las partes (a) y (b) de la figura 25, respectivamente. Los resultados incluidos en la parte (a) de esta figura son los mismos que los de la figura 21(c). Notar también que de acuerdo con (66), el valor crítico del factor de convergencia es $\mu_{max} = 2/\lambda_{max} = 1.1$, que es un poco mayor que el usado en la figura 25(b) ($\mu = 1.0$).

Los resultados para $\mu = 0.3$ y $mu = 1.0$ en términos de $w_1(n)$ y $w_2(n)$ se muestran en las partes (a) y (b) de la figura 26. Aquí nuevamente los resultados incluidos en la parte (a) de la figura son los mismos que los de la figura 23(c).

Observaciones

Con base en los resultados de los experimentos 1 y 2 es posible incluir las siguientes observaciones:

1. La trayectoria $[\nu_1(n), \nu_2(n)]$ con n como parámetro es normal al lugar geométrico de $[\nu_1(n), \nu_2(n)]$ para $J(n)$ fijo. Esto se aplica también para la trayectoria de $[w_1(n), w_2(n)]$ para $J(n)$ fijo.
2. Cuando los autovalores son iguales, la trayectoria $[\nu_1(n), \nu_2(n)]$ ó $[w_1(n), w_2(n)]$, con n como parámetro, es una línea recta. Esto se ilustra en las figuras 20(a) ó 22(a).
3. Cuando las condiciones son adecuadas para que el valor inicial de $\nu(0)$ esté sobre el eje ν_1 ó el eje ν_2 , la trayectoria $[\nu_1(n), \nu_2(n)]$, con n como parámetro, la trayectoria es una línea recta. Esto se ilustra en la figura 20(b), donde $\nu_1(0)$ es aproximadamente cero. En consecuencia, la trayectoria de $[w_1(n), w_2(n)]$, con n como parámetro, es también una línea recta como ilustrado en la figura 22(b).
4. Excepto por dos casos especiales ((1) autovalores iguales y (2) condiciones iniciales adecuadas), la trayectoria $[\nu_1(n), \nu_2(n)]$, con n como parámetro, la trayectoria sigue un camino curvo como ilustrado en la figura 21(c). En consecuencia, la trayectoria de $[w_1(n), w_2(n)]$, con n como parámetro, sigue también un camino curvo como ilustrado en la figura 23(c). Cuando la dispersión de autovalores es muy alta, pueden suceder dos cosas:
 - La superficie de error toma la forma de un valle pronunciado.
 - Las trayectorias de $[\nu_1(n), \nu_2(n)]$ y de $[w_1(n), w_2(n)]$ desarrollan curvaturas diferentes. Ambas consideraciones se ilustran

en las figuras 21(d) y 23(d), respectivamente, para el caso de $\chi(\mathbf{R}) = 100$.

5. El algoritmo SD converge más rápido cuando los autovalores λ_1 y λ_2 son iguales o cuando la condición inicial es adecuada, en cuyos casos la trayectoria formada siguiendo los puntos $\nu(0), \nu(1), \nu(2), \dots$ es una línea recta, el camino más corto.
6. Para un factor de convergencia μ , si la dispersión de los autovalores $\chi(\mathbf{R})$ aumenta (o sea \mathbf{R} es peor condicionada) el lugar geométrico de $[\nu_1(n), \nu_2(n)]$ para valores fijos de $J(n)$ se vuelve cada vez más angosto y abrupto (o sea, el eje menor se vuelve más pequeño).
7. Cuando el factor de convergencia μ es pequeño, el comportamiento transitorio del algoritmo SD es **sobreamortiguado**, en el sentido que la trayectoria formada siguiendo $\nu(0), \nu(1), \nu(2), \dots$ sigue un camino continuo. Cuando, por otro lado, m toma el máximo valor permitido $2/\lambda_{max}$, el comportamiento transitorio del algoritmo SD es **subamortiguado**, en el sentido que la trayectoria exhibe oscilaciones. Estas formas de comportamiento transitorio diferentes se ilustran en las partes (a) y (b) de la figura 25 en términos de $\nu_1(n)$ y $\nu_2(n)$. Los resultados correspondientes en términos de $w_1(n)$ y $w_2(n)$ se muestran en las partes (a) y (b) de la figura 26

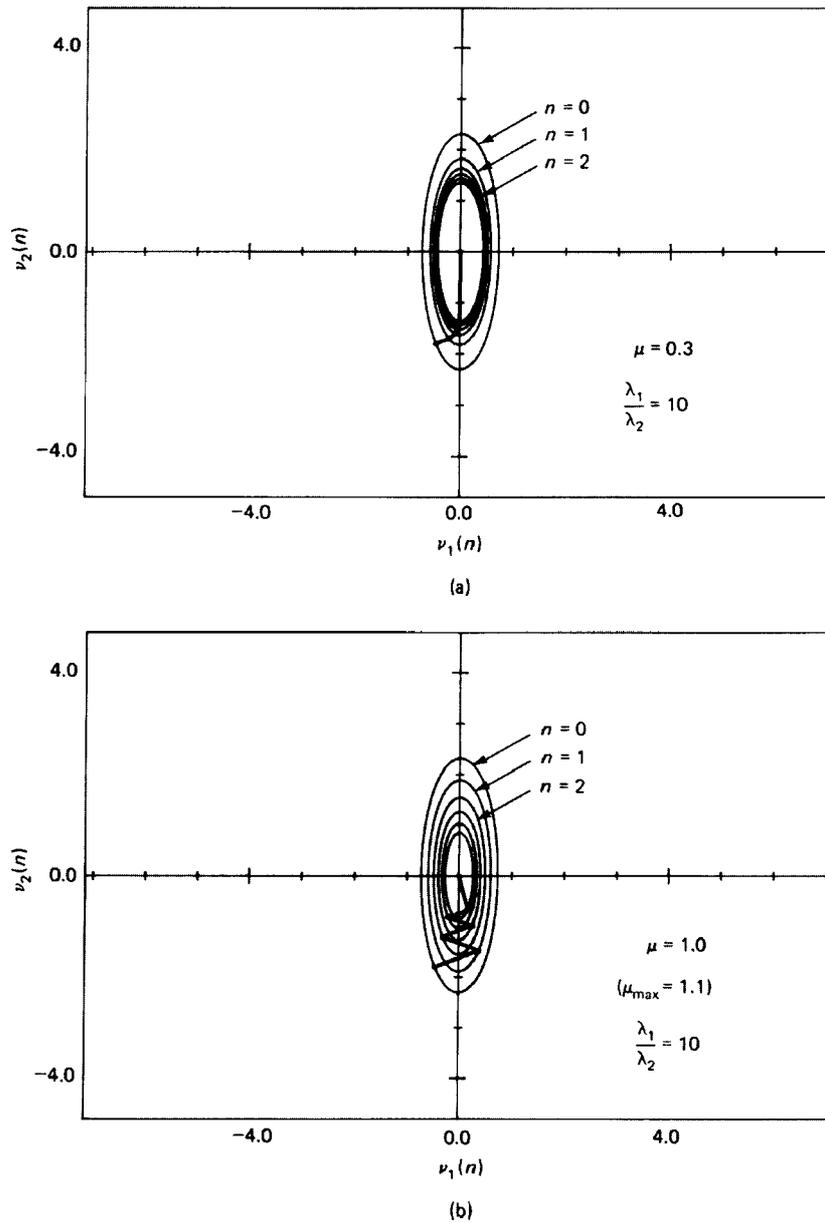


Figura 25: Lugar geométrico de $\nu_1(n)$ versus $\nu_2(n)$ para el algoritmo SD con dispersión de autovalores $\chi(\mathbf{R}) = 10$ y factor de convergencia variable: a) sobreamortiguado $\mu = 0.3$, b) subamortiguado, $\mu = 1.0$.

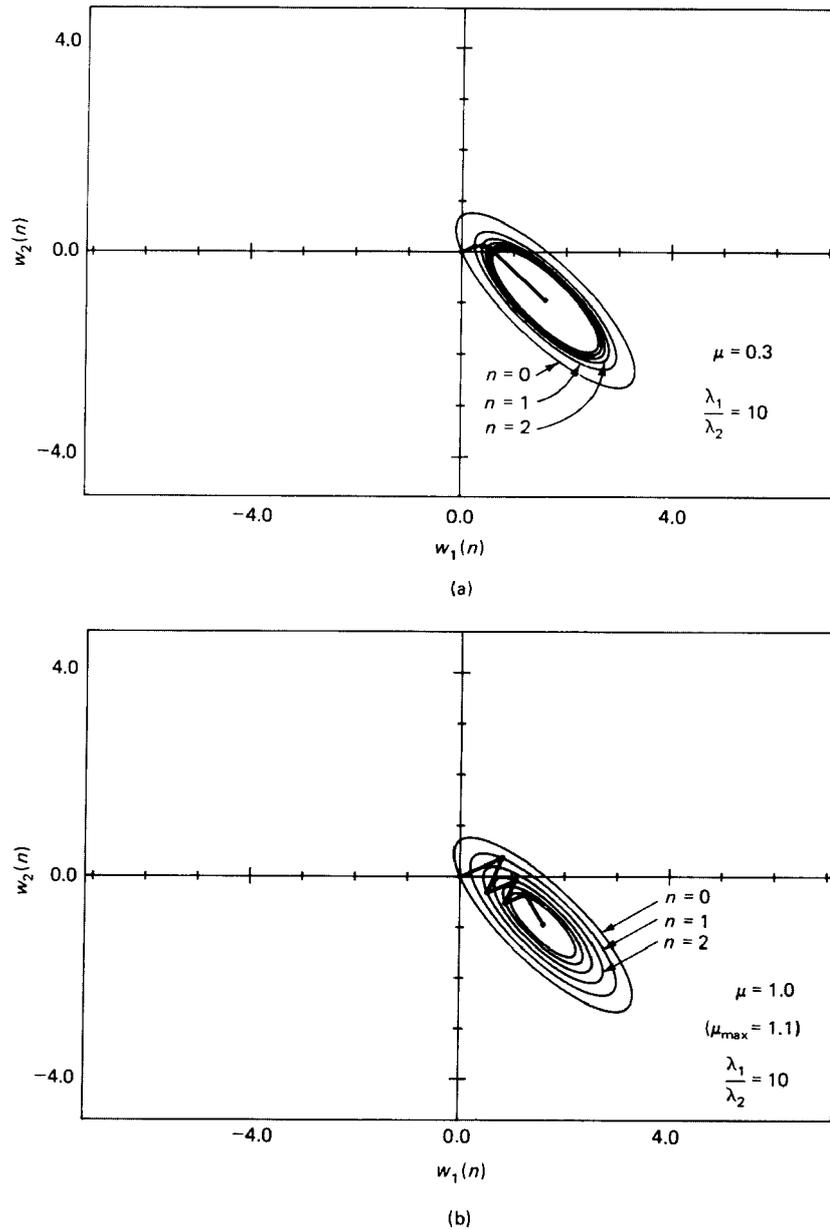


Figura 26: Lugar geométrico de $w_1(n)$ versus $w_2(n)$ para el algoritmo SD con dispersión de autovalores $\chi(\mathbf{R}) = 10$ y factor de convergencia variable: a) sobreamortiguado $\mu = 0.3$, b) subamortiguado, $\mu = 1.0$.