NUMERISKA ALGORITMER FÖR LÖSNING AV MINIMALTIDSPROBLEMET

KRISTER MÅRTENSSON

Rapport RE-5 okt. 1966

NUMERISKA ALGORITMER FÖR LÖSNING AV MINIMALTIDSPROBLEMET

Examensarbete

av

Krister Mårtensson

INNEHÅLLSFÖRTECKNING

1. FORMULERING AV PROBLEMET

Betrakta en punkt $x = (x_1, x_2 \ldots x_n)$ i ett n-dimensionellt tillståndsrum. För dess rörelse antages följande system av differentialekvationer gälla

$$\frac{dx_i}{dt} = f_i(x_1, \ldots x_n, u_1, \ldots u_r, t) \qquad i = 1, 2, \ldots n \qquad (1.1)$$

$u = (u_1, u_2 \ldots u_r)$ är en r-dimensionell kontrollvektor med vilken man kan göra ingrepp i systemet. Om det för systemet är givet en styrlag $u(t) = (u_1(t), u_2(t), \ldots u_r(t))$ och ett begynnelsetillstånd $x(0) = (x_1(0), x_2(0), \ldots x_n(0))$, bestämmer (1.1) entydigt punktens läge och rörelse i tillståndsrummet för varje tidpunkt t. I regel är f_i olinjära funktioner i x och u, men för det viktiga specialfall att f_i är linjära och systemet autonomt, dvs tiden t ingår ej explicit i f_i, kan (1.1) skrivas

$$\frac{dx_i}{dt} = \sum_{r=1}^{n} a_i^r x_r + \sum_{\delta=1}^{r} b_i^\delta u_\delta \qquad i = 1 \ldots n \qquad (1.2)$$

eller i matrisform

$$\frac{dx}{dt} = Ax + Bu \qquad (1.3)$$

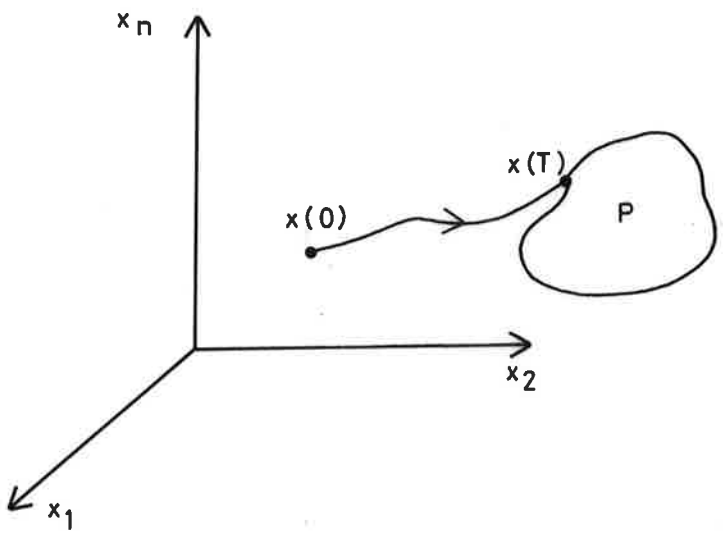
A och B är matriser av dimension n x n resp. n x r. En stor del av optimeringsteorins resultat och metoder är begränsade till det linjära fallet, och det kommer i fortsättningen att antagas att systemet kan skrivas på formen (1.2) och (1.3)

Kontrollvektorn u(t) antages styckvis kontinuerlig och styckvis kontinuerligt deriverbar. Att systemet är autonomt innebär att ett styringrepp vid tiden t_1 ger samma ändring i systemet som det skulle göra vid tiden t_2, förutsatt att systemet befinner sig i samma tillstånd

vid de båda tidpunkterna. Man kan alltså lämpligen sätta tiden t = 0 vid början av ett styringrepp vars verkningar man vill undersöka. Rent fysikaliskt är vidare att vänta att vissa begränsningar finnes på u. I ett mekaniskt system är ju exempelvis oändligt stora krafter en orimlighet. Vi ställer därför kravet att u(t) i varje ögonblick måste tillhöra någon begränsad mängd U. U är någon delmängd i den rymd som spännes upp av $u_1, u_2, \ldots, u_r$. U kan till exempel vara en sfär $\sum_{i=1}^{r} u_i^2 \leq 1$ dvs styrsignalen är begränsad i normen, eller U kan vara det inre av en kub $|u_i| < 1 \quad i = 1,2 \ldots r$, dvs varje komponent i u är begränsad till att ligga mellan -1 och 1. I fortsättningen antages det senare, som i det fall att u blott har en komponent reduceras till $|u| < 1$. En styrsignal som tillhör U säges vara tillåten eller admissibel. Syntesproblemet är emellertid i regel inte lösbart om U är en öppen mängd, och vi lägger det ytterligare kravet på U att det skall vara en kompakt mängd, dvs $|u| \leq 1$. Längre fram kommer det att visa sig att kompaktheten hos U inte är ett tillräckligt krav, utan U måste dessutom vara konvex och innehålla origo som inre punkt. I fallet $|u| \leq 1$ är detta uppfyllt.

Naturligt vore att på samma sätt införa restriktioner på tillståndsvariablerna, dvs införa någon delmängd X i tillståndsrummet med krav att för alla t $x(t) \in X$. Fysikaliskt kan det till exempel vara skäl att begränsa hastigheter eller accelerationer i ett system. Det skulle emellertid föra utanför ramen för examensarbetet, och vi antar därför att $x_i(t)$ kan antaga godtyckliga värden.

Problemet är nu följande. Antag att systemets initialtillstånd x(0) och någon delmängd P i tillståndsrummet är givna. Hur skall man välja sin styrsignal för att systemet på kortast möjliga tid skall transformeras från x(0) till någon punkt som tillhör P?



Denna minimaltid betecknas med T och antas i det allmänna fallet inte given. x(T) är inte på förhand bestämd utom för det fall att P urartar till en punkt, och det gäller alltså då att överföra systemet till ett bestämt tillstånd på kortast möjliga tid.

Minimaltidsproblemet är ett specialfall av ett mera generellt problem, nämligen att överföra systemet från x(0) till P på ett sådant sätt att någon given funktional S antar ett minimalt (eller maximalt) värde. S ges i regel som en integral

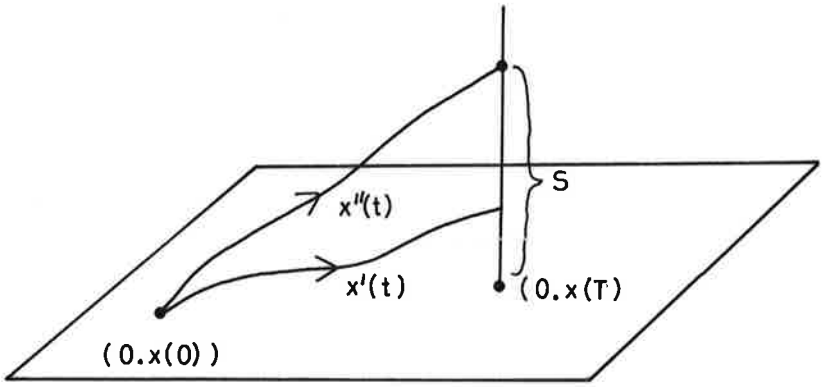
$$S = \int_0^T F(x_1, \ldots x_n, u_1 \ldots u_r, t)\, dt \qquad (1.4)$$

där F vanligtvis är någon positiv skalär funktion i x, u eller t. Om till exempel styrsignalen u i något system betecknar bränsletillförsel, och F sättes lika med u^2, innebär minimering av S en transformation från x(0) till P med så lite bränsleåtgång som möjligt.

Inför ytterligare en tillståndsvariabel $x_o(t)$ och sätt

$$x_o(t) = \int_0^t F(x_1 \ldots x_n, u_1 \ldots u_r, t)\, dt \qquad (1.5)$$

Denna ekvation adderas till systemekvationerna (1.1) med begynnelsevillkoret $x_0(0) = 0$. Problemet har då överförts till att välja en styrsignal u(t) så att systemet överföres från begynnelsetillståndet (0,x(0)) till en rät linje genom punkten (0,x(T)). Denna linje är parallell med x_0-axeln och det gäller att minimera den nollte tillståndsvariabeln.



Sättes nu F ≡ 1 får man S = T och $x_0(T) = T$. Har man funnit en styrsignal som överför systemet till någon punkt på linjen på så sätt att $x_0(T)$ är så liten som möjligt har man alltså funnit en lösning på problemet att överföra systemet från x(0) till x(T) på kortast möjliga tid. I fortsättningen kommer enbart detta fall, det vill säga F ≡ 1 att behandlas.

Lösningen till minimaltidsproblemet kan fås med en del olika metoder. Meningen är här att använda Pontryagins maximiprincip och att utveckla en algoritm för numerisk lösning av problemet. Dynamisk programmering vore ett alternativ. Genom att transformera styrvariabeln kan dock i vissa fall den klassiska variationskalkylen användas. Om t.ex. $U : \{ u, |u| \leq 1 \}$ kan en ny styrvariabel v införas så att $u = \sin v$ där v nu tillhör den öppna mängden $(-\infty,\infty)$. I nästa avsnitt kommer kortfattat att redogöras för maximiprincipen och några egenskaper hos optimal styrning.

2. MAXIMIPRINCIPEN OCH EGENSKAPER HOS OPTIMAL STYRNING

Betrakta åter grundsystemet (1.1) nu utökat med den extra tillståndsvariabeln x_0

$$\frac{dx_i}{dt} = f_i(x,u) \qquad i = 0, 1 \ldots n \tag{2.1}$$

Systemets begynnelsevärde antages givet $x(0) = (0, x_1(0) \ldots x_n(0))$. På samma sätt som förut kan (2.1) i det linjära fallet skrivas i matrisform

$$\frac{dx}{dt} = Ax + Bu \tag{2.2}$$

där A nu har dimension $(n + 1) \times (n + 1)$ och B $(n + 1) \times r$. f_i antages kontinuerliga och kontinuerligt deriverbara i $x_0, x_1 \ldots x_n$, dvs $\frac{\partial f_i}{\partial x_\alpha}$ existerar och är kontinuerliga för alla i och α. I det linjära fallet är detta villkor uppfyllt. Inför en ny vektor $p(t) = (p_0(t), p_1(t) \ldots p_n(t))$, vars komponenter $p_i(t)$ satisfierar ekvationerna

$$\frac{dp_i}{dt} = -\sum_{\alpha=0}^{n} \frac{\partial f_\alpha(x,u)}{\partial x_i} p_\alpha \qquad i = 0, 1 \ldots n \tag{2.3}$$

$p(t) = (p_0(t), p_1(t) \ldots p_n(t))$ säges vara den till x(t) adjungerade vektorn. Fastlägges begynnelsevärdet $p(0) = (a_0, a_1, a_2 \ldots a_n)$ bestämmer (2.3) för varje styrsignal u(t) entydigt p(t). I det linjära fallet kan (2.3) skrivas

$$\frac{dp}{dt} = -A^T p \tag{2.4}$$

p(t) är alltså i detta fall oberoende av u(t), och är bestämda så snart initialvärdet p(0) fastlagts.

För mekaniska system beskriver (2.3) och (2.4) systemets impuls, varför p(t) i fortsättningen även kommer att benämnas impulsvektor. Vi bildar nu Hamiltonfunktionen

$$\mathcal{H}(x,p,u) = \sum_{i=0}^{n} p_i f_i(x,u) \qquad (2.5)$$

som är skalärprodukten av vektorerna p(t) och $\frac{dx}{dt}$ eller alternativt projiceringen av $\frac{dx}{dt}$ på p(t). Med hjälp av $\mathcal{H}$ kan (2.1) och (2.3) skrivas

$$\begin{aligned} \frac{dx_i}{dt} &= \frac{\partial \mathcal{H}}{\partial p_i} \qquad i = 0, 1 \ldots n \\ \frac{dp_i}{dt} &= -\frac{\partial \mathcal{H}}{\partial x_i} \qquad i = 0, 1 \ldots n \end{aligned} \qquad (2.6)$$

Om x och p fixeras är $\mathcal{H}$ en funktion av styrsignalen u, med definitionsområdet u ε U. Det största värde $\mathcal{H}$ kan antaga för ett visst val av x och p betecknas $\mathcal{M}(x,p)$

$$\mathcal{M}(x,p) = \sup_{u \varepsilon U} \mathcal{H}(x,p,u) \qquad (2.7)$$

Betrakta speciellt det tidsoptimala problemet, där alltså $f_0 = 1$. Hamiltonfunktionen reduceras då till

$$\mathcal{H}(x,p,u) = p_0 + \sum_{i=1}^{n} p_i f_i(x,u) \qquad (2.8)$$

Ur (2.3) fås vidare $\frac{dp_0}{dt} = 0$, dvs p_0 = konstant.

Termen $p_0 f_0$ är alltså utan intresse när det gäller att söka maximum av $\mathcal{H}$, då den för alla t är konstant = p_0. Resterande termer i Hamiltonfunktionen betecknas med H(x,p,u)

$$H(x,p,u) = \sum_{i=1}^{n} p_i f_i(x,u) \qquad (2.9)$$

och maximum av H med M(x,p)

$$M(x,p) = \sup_{u \in U} H(x,p,u) \qquad (2.10)$$

Vi kan nu formulera Pontryagins maximiprincip (se {1}).

Låt u(t), $0 \leq t \leq T$ vara en admissibel styrsignal som överför systemet från x(0) till x(T), och x(t) den därtill hörande trajektorian. Ett nödvändigt villkor för att u(t) skall vara en tidsoptimal styrsignal, är att det existerar en till x(t) och u(t) hörande vektor p(t), ej identiskt noll, så att

A. för alla t, $0 \leq t \leq T$ väljes $u \in U$ så att H(x,p,u) antar sitt maximum M(x,p)

B. vid sluttidpunkten gäller $M(x(T), p(T)) \geq 0$.

Speciellt för linjära, autonoma system, dvs t ingår ej explicit i $\mathcal{H}$ gäller $\mathcal{B}$ för varje tidpunkt t, $0 \leq t \leq T$, ty

$$\frac{\partial \mathcal{H}}{\partial t} = \mathcal{H}_t + \mathcal{H}_x \cdot \frac{dx}{dt} + \mathcal{H}_p \cdot \frac{dp}{dt} = \mathcal{H}_t = 0$$

Observera att maximiprincipen i allmänhet bara ger de nödvändiga villkoren för optimal styrning. För det linjära fallet (2.2) och för något mera allmänna system beskrivna av

$$\frac{dx_i}{dt} = \sum_{\alpha=1}^{n} a_{i\alpha}(t)\, x_\alpha + \phi_i(u_1 \ldots u_r) \qquad i = 1 \ldots n \qquad (2.11)$$

gäller emellertid att maximivillkoret är ett "nästan" tillräckligt krav (se {2}).

Observera vidare att maximiprincipen inte ger någon explicit lösning på problemet att på kortast möjliga tid föra systemet från x(0) till x(T), utan säger endast vilka villkor en tidsoptimal lösning, vilken som helst, måste uppfylla. Beroende på initialvärdena på den adjungerade vektorn kommer systemet att följa skilda trajektorier ut från punkten x(0). Dessa är visserligen i och för sig optimala under förutsättning att man styrt systemet enligt maximiprincipen, men det är fortfarande inte givet att någon, och i så fall en entydigt bestämd trajektoria går genom x(T). Maximiprincipen kan alltså sägas uttrycka hur man i varje ögonblick skall styra systemet, snarare än hur man skall styra för att nå en viss punkt i tillståndsrummet. Detta kan också uttryckas på följande sätt. Antag att man gjort ett visst val av p(0) = a. På tiden t'' transformeras systemet av en optimal styrsignal u(t) till punkten x'' och systemet passerar vid tiden $t' < t''$ punkten x'. u(t) är alltså en tidsoptimal styrsignal då det gäller att överföra systemet från x(0) till x'', men den är även optimal då det gäller att träffa x'. En tidsoptimal styrning kan alltså sägas vara "ständigt optimal", och inte optimal endast för det sluttillstånd som åsyftas.

Beträffande entydigheten och existensen av en optimal lösning, se Pontryagin {1}. Vi nöjer oss här med att fastslå att i det linjära fallet är entydigheten och existensen av en optimal lösning tryggad så snart det existerar någon styrlag u(t) som transformerar systemet från x(0) till x(T). I det linjära fallet kan Hamiltonfunktionen skrivas

$$H(x,p,u) = \langle p \mid Ax \rangle + \langle p \mid Bu \rangle \qquad (2.12)$$

Eftersom u bara ingår i den senare termen reduceras problemet till att söka maximum av $\langle p \mid Bu \rangle$.

Detta maximum betecknas P(p)

$$P(p) = \sup_{u \in U} < p \mid Bu > \qquad (2.13)$$

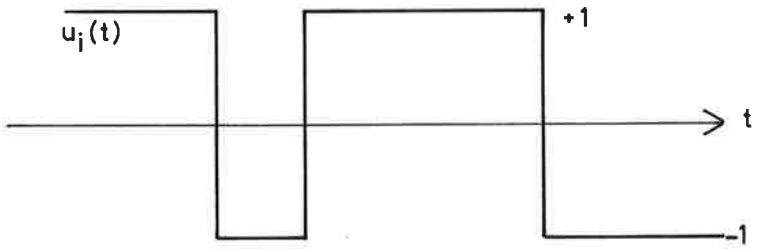
eller

$$P(p) = \sup_{u \in U} \sum_{i=1}^{r} \left(\sum_{j=1}^{n} p_j \cdot b_{ij} \right) u_i$$

Om vi nu förutsätter att U är kuben $| u_i | \leq 1 \quad i = 1 \ldots n$, fås tydligen supremum då

$$u_i = \operatorname{sign} \sum_{j=1}^{n} p_j \cdot b_{ij} \qquad (2.14)$$

det vill säga styrsignalen u(t) ligger alltid i något av kubens hörn. $u_i(t)$ är alltså styckvis konstanta och antar endast värdena +1 och -1 (bang-bang).



I specialfallet r = 1, och under förutsättning att samtliga koefficienter b_i utom en är noll (kan alltid fås genom transforamtion av systemet) får man

$$u(t) = \operatorname{sign} (b_i p_i(t)) \qquad (2.15)$$

Man har nu vissa möjligheter att med hjälp av matrisen A avgöra hur många teckenväxlingar u(t) kan göra (se {1}). Är sålunda samtliga egenvärden till A reella följer ur (2.4) att $p_i(t)$ högst har n intervall med konstant tecken, det vill säga u(t) skiftar tecken högst n-1 gånger. Är å andra sidan egenvärden komplexa blir $p_i(t)$ periodisk och det finns ingen övre gräns för antalet teckenväxlingar hos u(t).

Antag nu speciellt att det önskade slutţillståndet är origo. Inför en mängd C(t) som mängden av alla tillstånd som kan transformeras till sluttillståndet på tiden t med en godtycklig styrsignal (ej nödvändigtvis uppfyllande maximiprincipen). Pontryagin {1} visar att under förutsättning att det för varje x existerar en optimal styrning som överför x till origo, så är mängden C(t) konvex, sluten och innehåller inre punkter.

Vänd nu på problemet och antag att C(t) är de punkter som från begynnelsetillståndet kan nås på tiden t med en godtycklig styrsignal. Begynnelsetillståndet anses vara godtyckligt x(0), ej nödvändigtvis origo. Det är nu lätt att på samma sätt som i {1} visa att C(t) är konvex. Antag nämligen $u_1(t)$ och $u_2(t)$ är admissibla kontroller som på tiden t överför systemet i x_1 resp x_2. Det gäller alltså

$$e^{At} \cdot \{ x(0) + \int_0^t e^{-As} \cdot B \cdot u_1(s)\, ds \} = x_1$$

och

$$e^{At} \cdot \{ x(0) + \int_0^t e^{-As} \cdot B \cdot u_2(s)\, ds \} = x_2 \qquad (2.16)$$

Om α_1 och α_2 är godtyckliga positiva tal som uppfyller $\alpha_1 + \alpha_2 = 1$ så ligger $x_3 = \alpha_1 x_1 + \alpha_2 x_2$ på den linje som förbinder x_1 och x_2, och $u_3 = \alpha_1 u_1 + \alpha_2 u_2$ på en linje mellan u_1 och u_2. u_3 är alltså en admissibel kontroll eftersom U är konvex.

Multiplicera (2.16) med α_1 resp α_2 och summera

$$e^{At} \{ x(0) + \int_0^t e^{-As} \cdot B \cdot (\alpha_1 u_1(s) + \alpha_2 u_2(s))\, ds \} =$$

$$= \alpha_1 x_1 + \alpha_2 x_2 = x_3 \qquad (2.17)$$

Men x_1 och x_2 är godtyckliga, varför det för alla par av punkter som tillhör C(t) måste gälla, att varje punkt på en rät linje mellan punkterna också tillhör C(t). C(t) är alltså även i detta fall konvex. Vidare är C(t) sluten, och randen till C(t) utgöres av de punkter man nått genom att styra optimalt. Antag nämligen att en sådan "optimalpunkt" x_1 är inre punkt i C(t). x_1 fås genom att integrera upp systemekvationerna och de adjungerade ekvationerna från noll till t. Antag att man fortsätter integrationen till tiden t + dt och därvid når x_2. x_2 är då en punkt som inte kan nås på kortare tid än t + dt. Men för dt tillräckligt liten ligger x_2 i en omgivning till x_1 som tillhör C(t), det vill säga x_2 tillhör C(t), och kan alltså nås på tiden t, vilket motsäger att x_2 inte kan nås på kortare tid än t + dt. Alltså utgöres randen till C(t) av "optimala punkter".

C(t) behöver inte nödvändigtvis innesluta begynnelsetillståndet x(0). Är emellertid x(0) sådant att Ax(0) = 0, orsakar styrsignalen $u(t) \equiv 0$ ingen ändring i systemet, och x(0) kommer då att för alla t vara en inre punkt i C(t). Det skall nu visas att impulsvektorn p(t) för alla t är den till C(t) utåtriktade normalen i x(t).

För att bevisa detta betraktas åter systemet (1.3)

$$\frac{dx}{dt} = Ax + Bu \tag{2.18}$$

med begynnelsetillståndet x(0).

De adjungerade variablerna bestämmes av

$$\frac{dp}{dt} = -A^T p \qquad p(0) = a \tag{2.19}$$

Maximiprincipen ger u = u(x,p), det vill säga (2.18) kan skrivas

$$\frac{dx_i}{dt} = f_i(x,p) \qquad i = 1 \ldots n \tag{2.20}$$

På samma sätt kan (2.19) skrivas

$$\frac{dp_i}{dt} = g_i(x,p) \qquad i = 1 \ldots n \tag{2.21}$$

Inför störningsmatrisen Y och Z

$$Y = \begin{bmatrix} \frac{dx_1}{da_1} & \frac{dx_1}{da_2} & \cdots & \frac{dx_1}{da_n} \\ \frac{dx_2}{da_1} & & & \vdots \\ \vdots & & & \\ \frac{dx_n}{da_1} & \cdots & & \frac{dx_n}{da_n} \end{bmatrix} \tag{2.22}$$

$$Z = \begin{bmatrix} \frac{dp_1}{da_1} & \frac{dp_1}{da_2} & \cdots & \frac{dp_1}{da_n} \\ \frac{dp_2}{da_1} & & & \vdots \\ \vdots & & & \\ \frac{dp_n}{da_1} & \cdots & & \frac{dp_n}{da_n} \end{bmatrix} \tag{2.23}$$

För Y och Z gäller följande differentialekvationer ({3})

$$\frac{dY}{dt} = f_x Y + f_p Z \qquad Y(0) = 0 \qquad (2.24)$$

och

$$\frac{dZ}{dt} = g_p Z + g_x Y \qquad Z(0) = I \qquad (2.25)$$

där f_x, f_p, g_p och g_x är matriserna

$$f_x = \begin{bmatrix} \frac{\partial f_1}{\partial f_2} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & & & \vdots \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} = A; \; f_p = \begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_1}{\partial p_2} & \cdots & \frac{\partial f_1}{\partial p_n} \\ \frac{\partial f_2}{\partial p_1} & & & \vdots \\ \vdots & & & \\ \frac{\partial f_n}{\partial P_1} & & \cdots & \frac{\partial f_n}{\partial p_n} \end{bmatrix} \qquad (2.26)$$

$$g_x = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & & & \vdots \\ \vdots & & & \\ \frac{\partial g_n}{\partial x_1} & & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \qquad g_p = \begin{bmatrix} \frac{\partial g_1}{\partial p_1} & \frac{\partial g_1}{\partial p_2} & \cdots & \frac{\partial g_1}{\partial p_n} \\ \frac{\partial g_2}{\partial p_1} & & & \\ & & & \\ \frac{\partial g_n}{\partial p_1} & & & \frac{\partial g_n}{\partial p_n} \end{bmatrix} = -A^T$$

Låt nu C(t) enligt ovan vara den slutna konvexa mängd av punkter som kan nås från x(0) på tiden t. Randen är de punkter som nås med en optimal styrning. Genom x(t) kan man då lägga ett nästan överallt entydigt tangentplan och vektorerna

$$\left(\frac{dx_1}{da_1}, \frac{dx_2}{da_1}, \ldots \frac{dx_n}{da_1}\right) = (Y_{11}, Y_{21}, \ldots Y_{n1}) \ldots \left(\frac{dx_1}{da_n}, \frac{dx_2}{da_n}, \ldots \frac{dx_n}{da_n}\right) =$$

$$= (Y_{1n}, Y_{2n}, \ldots Y_{nn})$$

är då vektorer i detta plan.

Betrakta skalärprodukterna

$$< (p_1(t), p_2(t), \ldots p_n(t)) \mid (Y_{1i}, Y_{2i}, \ldots Y_{ni}) > \qquad (2.27)$$

$$i = 1 \ldots n$$

Dessa utgör kolonnerna i radmatrisen p^TY. Bilda tidsderivatan $\frac{d}{dt}(p^TY)$

$$\frac{d}{dt}(p^TY) = \frac{d}{dt}(p^T) \cdot Y + p^T \frac{dY}{dt} = \left(\frac{dp}{dt}\right)^T \cdot Y + p^T \frac{dY}{dt} \qquad (2.28)$$

Men för $\left(\frac{dp}{dt}\right)^T$ gäller

$$\left(\frac{dp}{dt}\right)^T = -(A^Tp)^T = -p^TA \qquad (2.29)$$

Med (2.29) och (2.24) insatt i (2.28) fås

$$\frac{d}{dt}(p^TY) = -p^TAY + p^TAY + p^Tf_pZ = p^Tf_pZ \qquad (2.30)$$

Matrisen f_p kan skrivas

$$f_p = \begin{bmatrix} \frac{\partial f_1}{\partial u} \cdot \frac{\partial u}{\partial p_1} & \cdots & \frac{\partial f_1}{\partial u} \cdot \frac{\partial u}{\partial p_n} \\ \vdots & & \\ \frac{\partial f_n}{\partial u} \cdot \frac{\partial u}{\partial p_1} & \cdots & \frac{\partial f_n}{\partial u} \cdot \frac{\partial u}{\partial p_n} \end{bmatrix} = \begin{bmatrix} b_1 \cdot \frac{\partial u}{\partial p_1} & \cdots & b_1 \cdot \frac{\partial u}{\partial p_n} \\ \vdots & & \\ b_n \cdot \frac{\partial u}{\partial p_1} & \cdots & b_n \cdot \frac{\partial u}{\partial p_n} \end{bmatrix} \qquad (2.31)$$

där b_i är elementen i B. Antag nu samtliga b_i utom b_j lika med noll. (kan i det linjära fallet alltid fås genom en transformering av systemet). Samtliga rader i f_p utom rad j är då noll. Man får

$$p^T f_p = \left(p_j \cdot b_j \cdot \frac{\partial u}{\partial p_1} , p_j\, b_j \frac{\partial u}{\partial p_2} , \ldots, p_j\, b_j \frac{\partial u}{\partial p_n} \right) \qquad (2.32)$$

Maximiprincipen ger u = sign (p_j), och (2.32) kan skrivas

$$p^T f_p = \left(0, 0, \ldots, 0, p_j \cdot b_j \cdot \frac{du}{dp_j} , 0, \ldots 0 \right) \qquad (2.33)$$

Derivatan $\frac{du}{dp_j}$ är noll överallt utom i p_j:s nollgenomgångar.

$p_j \cdot \frac{du}{dp_j}$ är alltså en distribution. För störningsmatrisen Z gäller i det linjära fallet

$$\frac{dZ}{dt} = \left(-A^T \right) Z \qquad Z(0) = I \qquad (2.34)$$

Störningselementen z_{ij} är alltså kontinuerliga och ändliga för ändliga t, och termen $p^T f_p Z$ som konintegreras i distributionsteorins mening ger

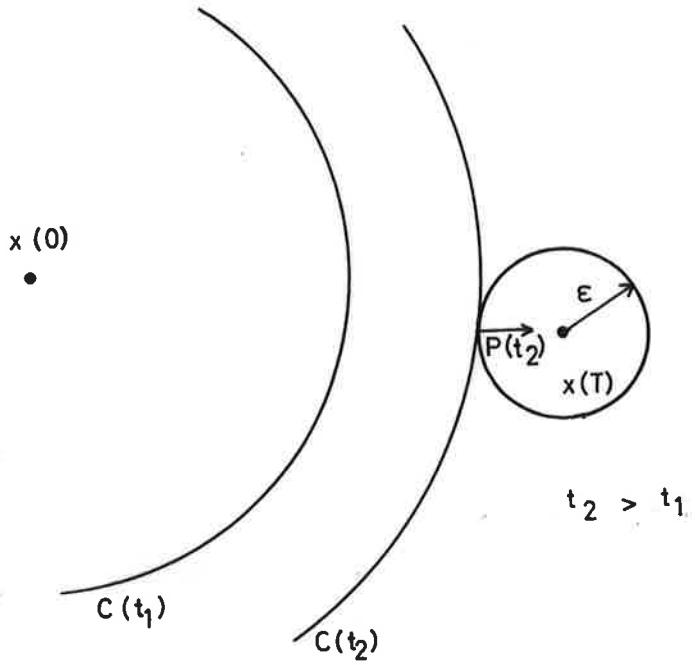
$$p^T(t) \cdot Y(t) = p^T(0) \cdot Y(0) \qquad (2.35)$$

Nu gäller $Y(0) = 0$, och sålunda

$$p^T(t)\, Y(t) = 0 \qquad (2.36)$$

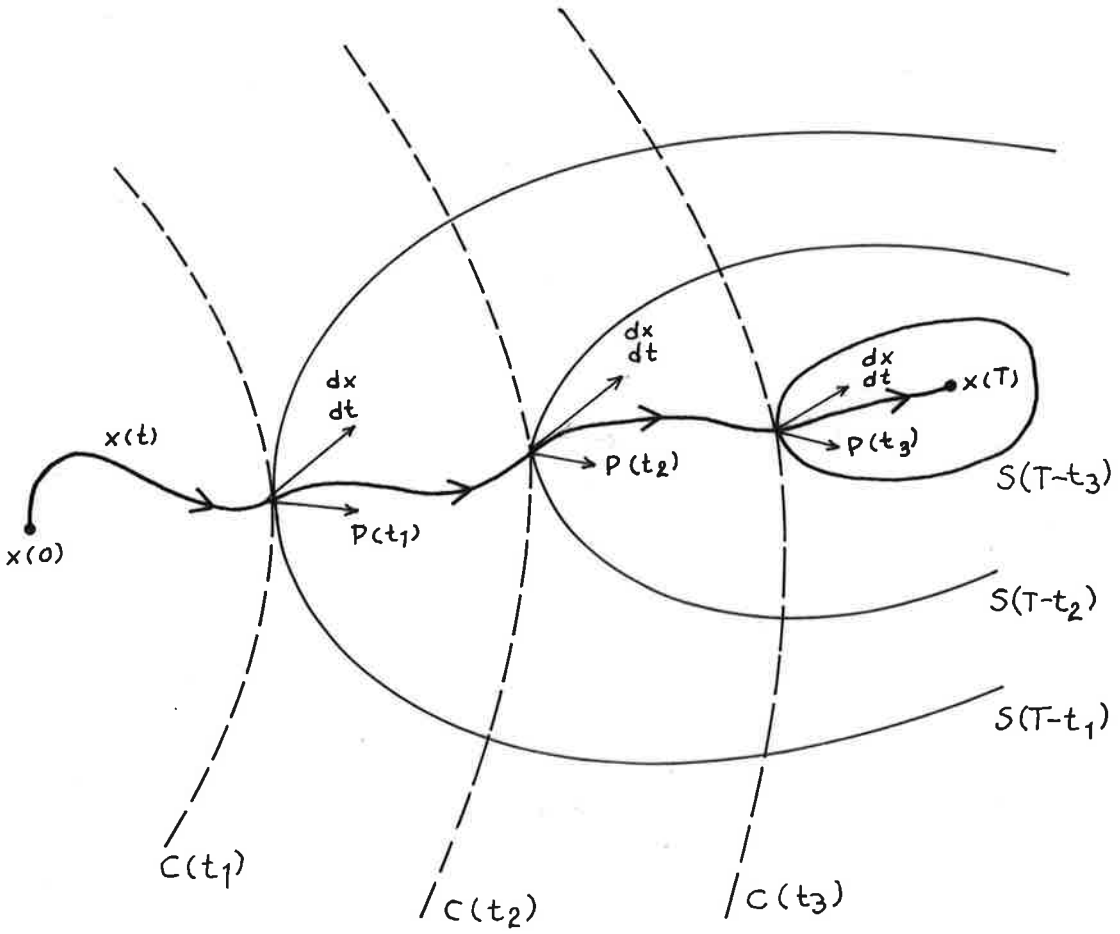
Därmed har det bevisats att p(t) är normal till C(t)-ytan för alla t. Att p(t) är utåtriktad följer slutligen ur maximiprincipen.

Detta resultat får nu en del intressanta konsekvenser när det gäller tidsoptimal styrning. Antag exempelvis att man släpper efter lite på kravet att exakt träffa sluttillståndet x(T), utan är nöjd om man blott kan nå ett sådant x så att $(x - x(T))^2 \leq \varepsilon^2$. Betrakta specialfallet n = 2, i vilket fall det alltså gäller att nå en cirkelformig omgivning till x(T). Det måste då gälla att den C(t)-kontur som representerar den minimala tiden tangerar denna cirkel, och p(t) måste vid sluttidpunkten vara normal inte bara till C(t) utan även till cirkeln.



Om x(T) inte är begränsad till en punkt utan exempelvis är något plan med normalriktningen $c = (c_1, c_2 \ldots c_n)$ måste på samma sätt gälla att $p(T) = (c_1, c_2, \ldots c_n)$. (eventuellt minustecken).

Det är nu möjligt att ge en direkt geometrisk teckning åt maximiprincipen. Beteckna som förut med C(t) de punkter som på tiden t kan nås från x(0), och beteckna med S(t) de punkter från vilka man på tiden t kan nå x(T). Vi förutsätter att det existerar en sökt minimaltid T. Med samma resonemang som ovan är det uppenbart att $C(t_1)$ tangerar $S(T - t_1)$, och $p(t_1)$ är följaktligen normal till båda dessa ytor i tangeringspunkten.



$t_3 > t_2 > t_1$

Om ytorna S(t) tillskrives en skalär funktion g(t) = t, lika med den minimala tid på vilken man kan nå x(T), blir alltså S iso-ytor till g och p har då gradientens riktning. Att försöka föra systemet i p-riktningen, det vill säga maximera $\mathcal{H} = < \frac{dx}{dt} \mid p >$, innebär då att man på så kort tid dt som möjligt söker nå någon innanför liggande yta S(t - Δt). Detta motiverar varför en optimal styrning kan sägas vara "ständigt optimal", och inte karakteristisk för speciella val av x(0) och x(T). Problemet återstår dock att finna det initialvärde för p(t) för vilket den optimala trajektorian går genom det avsedda sluttillståndet x(T).

Slutligen skall göras en kommentar beträffande adjungerade vektorn p(t). p(t) har definitionsmässigt införts som lösningen till systemet (2.3), i det linjära fallet (2.4), och visar sig sedan vara den utåtriktade normalen till ytor som kan nås på tiden t. Det är emellertid just ur ett sådant villkor som (2.3) och (2.4) kan härledas. Om man tänker sig ha funnit en optimal trajektoria x(t) som man på något sätt stör infinitesimalt, exempelvis i u(t), får man någon störning δ x(t) som om systemekvationerna är linjära uppfyller

$$\frac{d}{dt} (\delta x_i) = \sum_{\delta=1}^{n} \frac{\partial f_i}{\partial x_j} \delta x_j \qquad i = 1 \ldots n \qquad (2.37)$$

Sökes sedan en vektor p(t) sådan att

$$< \delta x(t) \mid p(t) > = \text{konstant} \qquad (2.38)$$

finner man att p(t) måste uppfylla (2.3) och (2.4). Se {1}.

3. NUMERISK LÖSNING AV MINIMALTIDSPROBLEMET

I detta avsnitt skall presenteras två olika metoder att numeriskt lösa minimaltidsproblemet. Med initialtillstånd x(0), sluttillstånd x(T) samt systemekvationerna (1.1) givna, är även ekvationerna för de adjungerade variablerna givna, och så snart initialvärdet $a = (a_1 \ldots a_n)$ fastlagts är styrsignalen entydigt bestämd ur maximiprincipen. Beroende på valet av a kommer systemtrajektoriorna att vid någon tidpunkt T passera olika punkter i fasrymden. Har man funnit ett a sådant att trajektorian går genom x(T) har man följaktligen bestämt den sökta optimala styrsignalen, och den sökta optimaltrajektorian. Här existerar emellertid en oklarhet, såtillvida att dessa begynnelsevärde är inte explicit givna av de övriga villkoren, utan man är hänvisad till ett sökande bland mängden av alla möjliga begynnelsevärden. Det gäller naturligtvis då att systematisera detta sökande, det vill säga skapa någon algoritm som från en i någon mening "vettig" första gissning på a, leder fram till de sökta begynnelsevärdena. Vad som menas med "vettig" gissning kommer att preciseras längre fram.

Två olika algoritmer skall här presenteras. Den första metoden härstammar från Levine {3}, och förutsätter i det skick densamme presenterat den inte någon apriori kännedom om den sökta minimaltiden T. Det kommer emellertid att visa sig att detta inte är helt korrekt, utan man är i de flesta fall tvungen att förutsätta tiden T känd för att metoden skall fungera. Detta kommer att motiveras och exemplificeras.

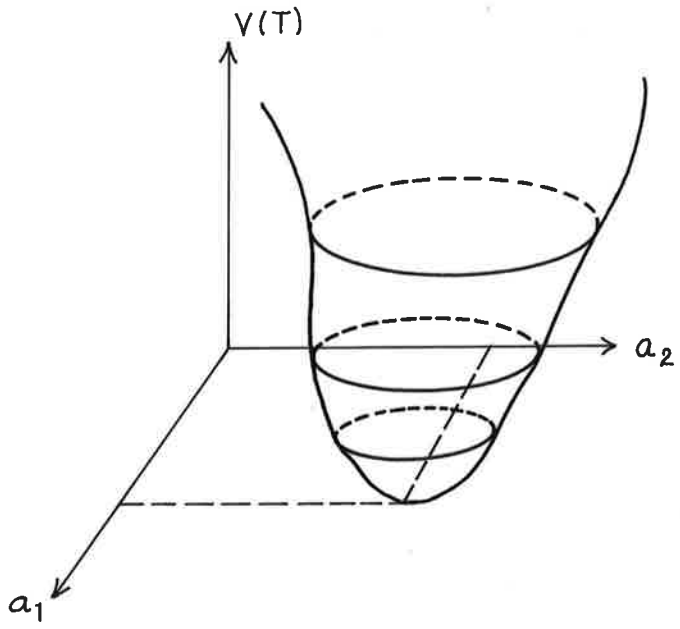
Metod två baserar sig på en metod av Neustadt {4}, där man använder sig av de i föregående avsnitt visade egenskaperna hos optimal styrning, och kan sägas vara en variant av denna metod. Den förutsätter inte någon kännedom om tiden T.

3.1 Algoritm för numerisk lösning av tidsoptimal styrning av känt T

Inför en skalär funktion V

$$V(t) = \sum_{i=1}^{n} (x_i^T - x_i(t))^2 \qquad (3.1)$$

För att undvika missförstånd betecknas nu sluttillståndet x^T, och är alltså endast i det optimala fallet lika med x(T) som vi låter beteckna systemets tillstånd vid tiden T. V som i det följande benämnes förlustfunktion, beror explicit av systemets tillstånd, men beror implicit av tiden och av begynnelsevärdet för adjungerade vektorn. V är alltid icke negativ och har minimum lika med noll då systemets tillstånd vid tiden T är lika med det avsedda sluttillståndet. Detta svarar mot ett speciellt val av $a = (a_1, a_2 \ldots a_n)$ och V:s implicita beroende av a kan därför åskådliggöras enligt figuren.



Antag nu en första gissning på a och integrera systemet fram till tiden T. Man når då en punkt på förlustytan, och med kännedom om a och någon egenskap hos ytan skall man då kunna bestämma något nytt a som ger mindre värde åt förlustfunktionen vid nästa integration. Om man känner gradienten $\frac{dV}{da}$ i varje punkt är uppenbarligen en enkel metod att åstadkomma denna minskning genom att ge a ett tillskott i negativa gradientriktningen (steepest descent) . För gradienten gäller

$$\frac{dV}{da}(T) = \frac{dV}{dx}(T) \cdot \frac{dx}{da}(T) \qquad (3.2)$$

där

$$\frac{dV}{da} = \left(\frac{dV}{da_1}, \frac{dV}{da_2}, \ldots \frac{dV}{da_n} \right), \quad \frac{dV}{dx} = \left(\frac{dV}{dx_1}, \frac{dV}{dx_2}, \ldots \frac{dV}{dx_n} \right)$$

och $\frac{dx}{da}$ den tidigare införda störningsmatrisen

$$\frac{dx}{da} = \begin{bmatrix} \frac{dx_1}{da_1} & \cdots & \frac{dx_1}{da_n} \\ & & \\ \frac{dx_n}{da_1} & & \frac{dx_n}{da_n} \end{bmatrix} = Y$$

På samma sätt införes åter störningsmatrisen

$$\frac{dp}{da} = \begin{bmatrix} \frac{dp_1}{da_1} & & \frac{dp_1}{da_n} \\ & & \\ \frac{dp_n}{da_1} & & \frac{dp_n}{da_n} \end{bmatrix} = Z$$

Y och Z satisfierar differentialekvationerna

$$\frac{dY}{dt} = f_x Y + f_p Z \qquad Y(0) = 0$$

$$\frac{dZ}{dt} = g_p Z + g_x Y \qquad Z(0) = I \qquad (3.3)$$

Då $\frac{dV}{dx}$ och $\frac{dx}{da}$ i varje tidpunkt är bestämda ur (3.1) resp. (3.3) är det alltså möjligt att för varje punkt på förlustytan bestämma gradienten. Initialvärdet för p(t) dateras upp enligt

$$a_{new} = a_{old} - k \cdot \frac{dV}{da}(T) \qquad (3.4)$$

där k är någon konstant. Under förutsättning att V inte har något annat minimum än det ovan förutsatta, garanterar (3.4) att man ständigt minskar förlustfunktionen och slutligen finner det a för vilket optimaltrajektorian passerar x^T.

Man kan nu skissera en algoritm.

1) Gissa ett värde på a och välj ett k.

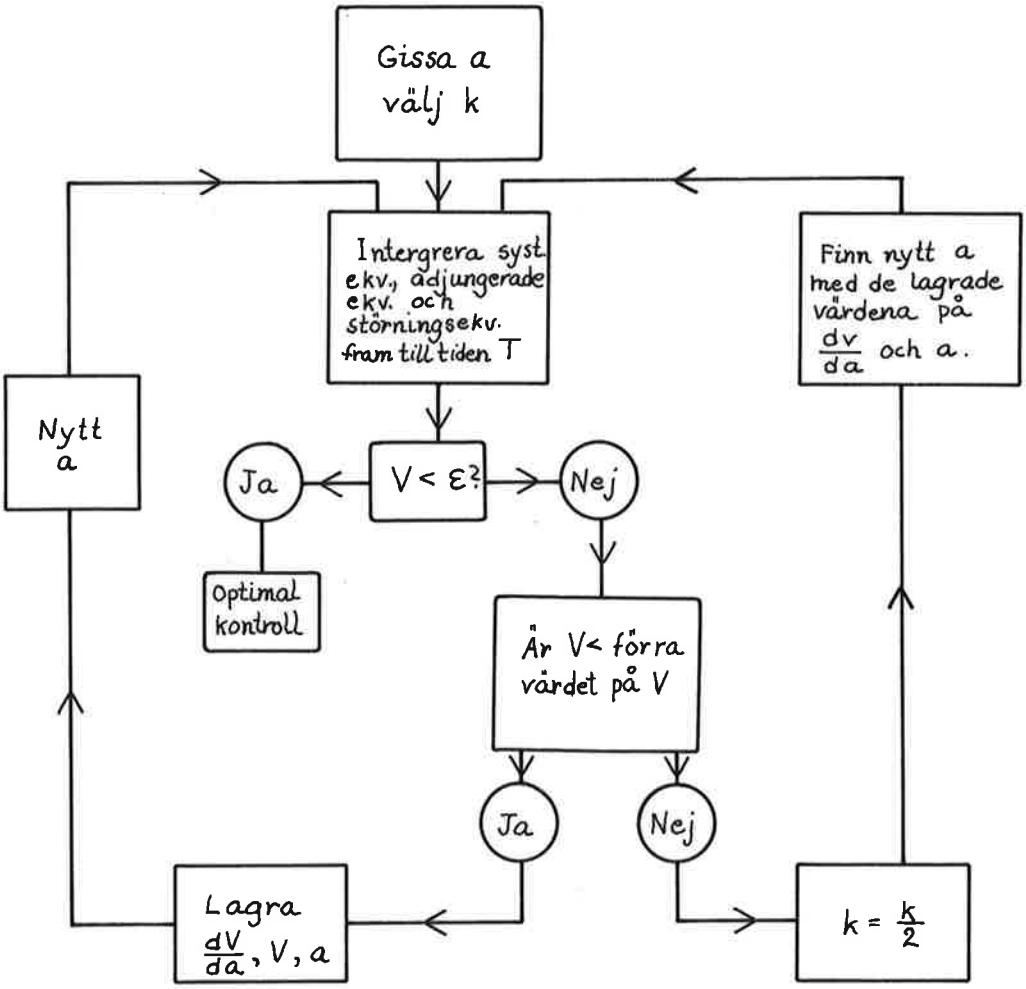
2) Integrera systemekvationerna, adjungerade ekvationerna och störningsekvationerna fram till tiden T.

3) Beräkna V(T). Om V(T) = 0 eller V(T) < ε är problemet löst.

4) Om V ≠ 0, undersök om det nya värdet på V är mindre än förra värdet på V. Om inte har man alltså passerat minimat och "gått för långt över på andra sidan". Halvera då k och använd det lagrade $\frac{dV}{da}$ till att finna ett nytt a och återvänd till 2. Detta upprepas tills det nya värdet på V är mindre än det lagrade.

5) Om 4 är uppfyllt beräkna $\frac{dV}{da}$ och uppdatera a. Återvänd till 2.

Steg 4 skulle kunna undvikas om man valde ett tillräckligt litet värde på k. Detta skulle emellertid kräva ett mycket större antal iterationer. 1-5 sammanfattas i ett flödesschema



Levine inför nu en gissad minimaltid t_f och gradienten $\frac{dV}{dt}(t_f)$. Genom att interera på t_f på samma sätt som för a det vill säga

$$t_{f\ new} = t_{f\ old} - k \cdot \frac{dV}{dt}(t_f) \qquad (3.5)$$

är meningen att man skall kunna finna även den sökta minimaltiden T. Detta kräver att $\frac{dV}{dt}$ är skild från noll utom för $t = T$ då $\frac{dV}{dt} = 0$. Detta är emellertid inte fallet. Inte ens för ett optimalt val av a kommer V att för vissa system blir en kontinuerligt avtagande funktion av tiden. V beräknas för varje punkt längs den trajektoria systemet beskriver, och det är då uppenbart att för en icke-optimal trajektoria som vid någon tidpunkt har ett minsta avstånd till x^T, $\frac{dV}{dt} = 0$ vid en tidpunkt som inte behöver vara den sökta minimaltiden. Om vi antar att iterationerna givit en sådan tidpunkt T' med $\frac{dV}{dt}(T') = 0$, kommer tiden att till nästa iteration vara oförändrad T', och problemet är då reducerat till att välja a så att V på denna tid blir så liten som möjligt. Man kan då inte förutsätta att det val av a som på tiden T' ger minimalt värde åt V är detsamma som ger minimalt värde åt V på tiden T, det vill säga det a som löser problemet. Förlustfunktionen V som funktion av a och t är alltså säkerligen inte någon enkel funktion med något entydigt minimum som man med "steepest descent" alltid garanteras falla ner i, utan kan säkerligen uppvisa en stor mängd singulära punkter. På två olika testexempel skall nu visas användning av den förslagna algoritmen med fixt T, och vad som kommer att hända om man försöker iterera även på tiden.

Testexempel

a) Dubbelintegrator

Givet systemekvationerna

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = u$$

sök styrsignalen u(t) så att systemet på minimal tid överföres från (1,0) till origo. T förutsättes känd = 2.0. Adjungerade ekvationerna

$$\frac{dp_1}{dt} = 0 \qquad p_1(0) = a_1$$

$$\frac{dp_2}{dt} = -p_1 \qquad p_2(0) = a_2$$

och man får alltså Hamiltonfunktionen

$$H = p_1 x_1 + p_2 u$$

u väljes så att H maximeras, det vill säga

$$u = \text{sign}\,(p_1)$$

Störningsekvationerna satisfierar enligt tidigare

$$\frac{dY}{dt} = f_x Y + f_p Z \qquad Y(0) = 0$$

$$\frac{dZ}{dt} = g_p Z + g_x Y \qquad Z(0) = I$$

eller utskrivet

$$\frac{d}{dt}\begin{bmatrix} \frac{dx_1}{da_1} & \frac{dx_1}{da_2} \\ \frac{dx_2}{da_1} & \frac{dx_2}{da_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{dx_1}{da_1} & \frac{dx_1}{da_2} \\ \frac{dx_2}{da_1} & \frac{dx_2}{da_2} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_1}{\partial p_2} \\ \frac{\partial f_2}{\partial p_1} & \frac{\partial f_2}{\partial p_2} \end{bmatrix} \begin{bmatrix} \frac{dp_1}{da_1} & \frac{dp_2}{da_1} \\ \frac{dp_1}{da_2} & \frac{dp_2}{da_2} \end{bmatrix}$$

och

$$\frac{d}{dt}\begin{bmatrix} \frac{dp_1}{da_1} & \frac{dp_1}{da_2} \\ \frac{dp_2}{da_1} & \frac{dp_2}{da_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial p_1} & \frac{\partial g_1}{\partial p_2} \\ \frac{\partial g_2}{\partial p_1} & \frac{\partial g_2}{\partial p_2} \end{bmatrix} \begin{bmatrix} \frac{dp_1}{da_1} & \frac{dp_1}{da_2} \\ \frac{dp_2}{da_1} & \frac{dp_2}{da_2} \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{dx_1}{da_1} & \frac{dx_1}{da_2} \\ \frac{dx_2}{da_1} & \frac{dx_2}{da_2} \end{bmatrix}$$

I de numeriska räkningarna approximeras $\frac{\partial f_2}{\partial p_2}$ med $(u(t) - u(t-h)) / (p(t) - p(t-h))$ där h är integrationslängden. Störningskomponenterna satisfierar alltså

$$\frac{dY_{11}}{dt} = Y_{21} \qquad Y_{11}(0) = 0$$

$$\frac{dY_{12}}{dt} = Y_{22} \qquad Y_{12}(0) = 0$$

$$\frac{dY_{21}}{dt} = (u(t) - u(t-h)) / (p(t) - p(t-h)) \cdot Z_{21} \qquad Y_{21}(0) = 0$$

$$\frac{dY_{22}}{dt} = (u(t) - u(t-h)) / (p(t) - p(t-h)) \cdot Z_{22} \qquad Y_{22}(0) = 0$$

samt

$$\frac{dZ_u}{dt} = 0 \qquad Z_{11}(0) = 1$$

$$\frac{dZ_{12}}{dt} = 0 \qquad Z_{12}(0) = 0$$

$$\frac{dZ_{21}}{dt} = -Z_{11} \qquad Z_{21}(0) = 0$$

$$\frac{dZ_{22}}{dt} = -Z_{12} \qquad Z_{22}(0) = 1$$

Förlustfunktion V valdes som

$$V = x_1^2(T) + x_2^2(T)$$

vilket ger

$$\frac{dV}{da} = \left(\frac{dV}{da_1}, \frac{dV}{da_2}\right) = \left(\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}\right) \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} =$$

$$= \{ 2x_1 Y_{11} + 2x_2 Y_{21},\ 2x_1 Y_{12} + 2x_2 Y_{22} \}$$

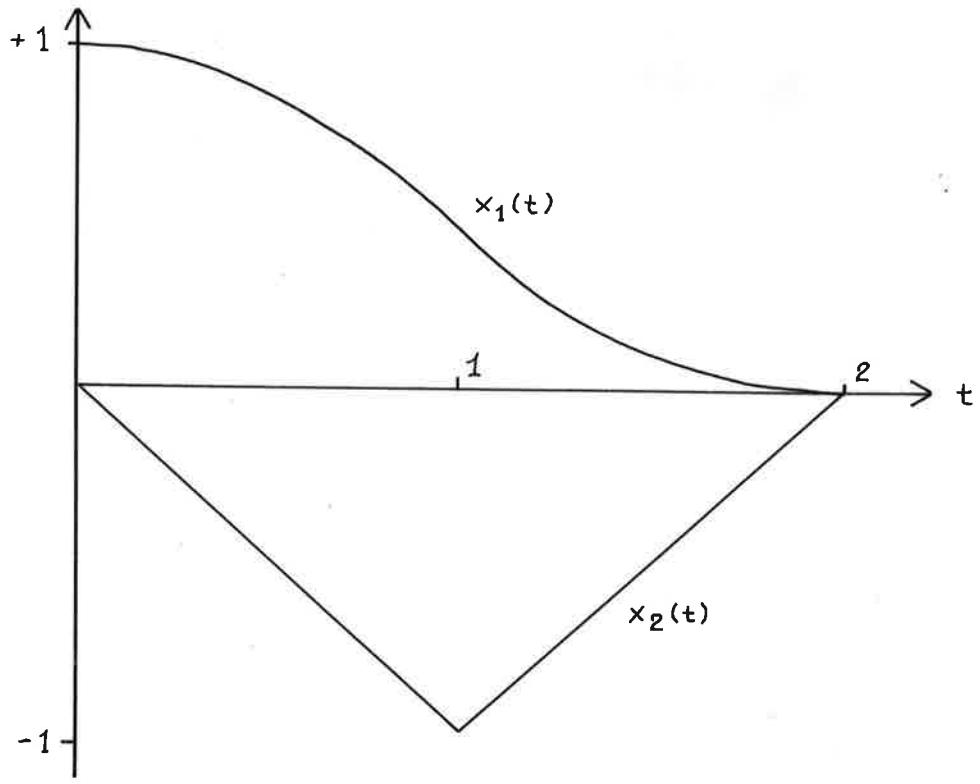
a_1 och a_2 skall alltså uppdateras enligt

$$a_{1\,new} = a_{1\,old} - 2k \cdot (x_1(T) \cdot Y_{11}(T) + x_2(T) \cdot Y_{21}(T))$$

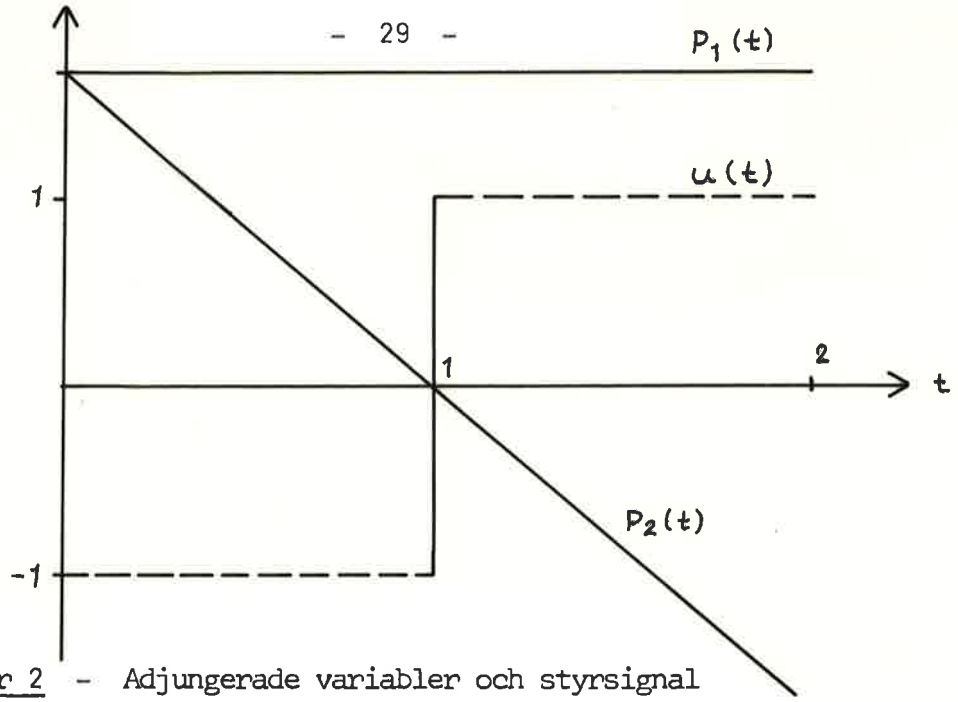
$$a_{2\,new} = a_{2\,old} - 2k \cdot (x_1(T) \cdot Y_{12}(T) + x_2(T) \cdot Y_{22}(T))$$

Problemet programmerades i ALGOL för körning på SMIL, och programmet återfinnes i appendix 1. Som integrationsrutin användes en fjärde ordningens Runge-Kutta med justering av integrationsintervallet då u byter tecken. Matrisen A har endast reella egenvärden, och som tidigare påpekats kan u då inte ändra tecken mer än en gång.

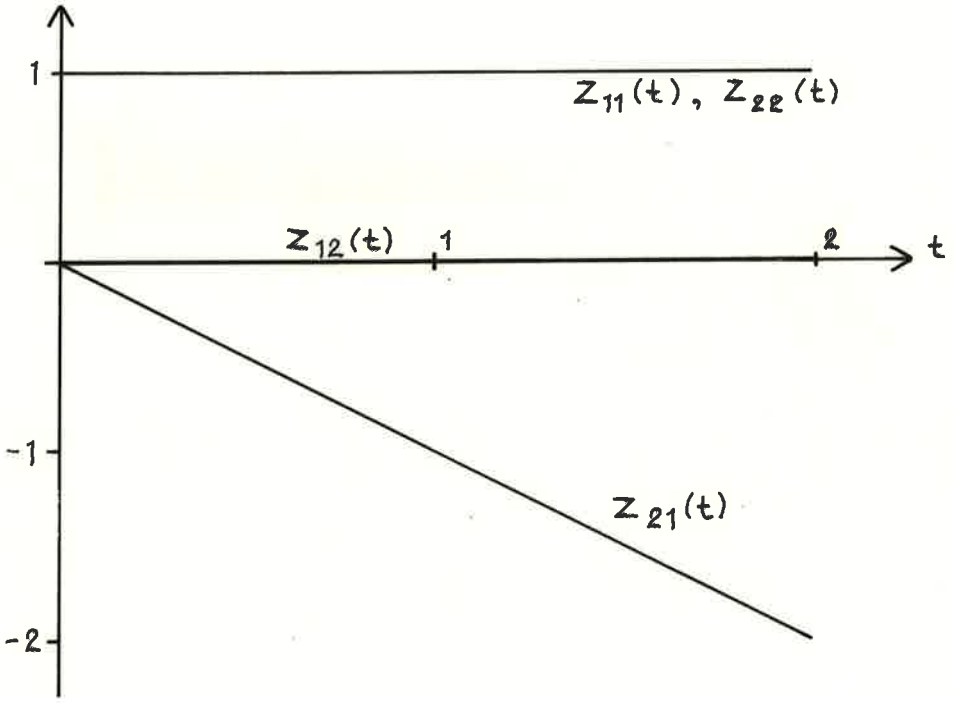
Problemet ansågs lösts då $V < \varepsilon$. Märk att i programmet har något oegentligt använts $u = -\text{sign}(p_2)$. Detta medför i det linjära fallet att adjungerade variablerna byter tecken medan tillståndsvariablerna blir oförändrade. Programmet testades för en del olika kombinationer på begynnelsevärdena a_1 och a_2 och visade inga svårigheter beträffande konvergensen mot optimalt a. Exempelvis med $a_1 = 1.0$, $a_2 = 0$, $k = 0.1$ och $\varepsilon = 0.005$ nåddes önskad noggrannhet redan efter tre iterationer. Det optimala värdet på a visade sig vara $a_1 = 1.63$ och $a_2 = 1.65$. Fig. 1 visar tillståndsvariablerna längs optimaltrajektorian, och fig. 2 adjungerade variablerna med den optimala styrsignalen. I fig. 3 har plottats störningsparametrarna Z och i fig. 4 Y.



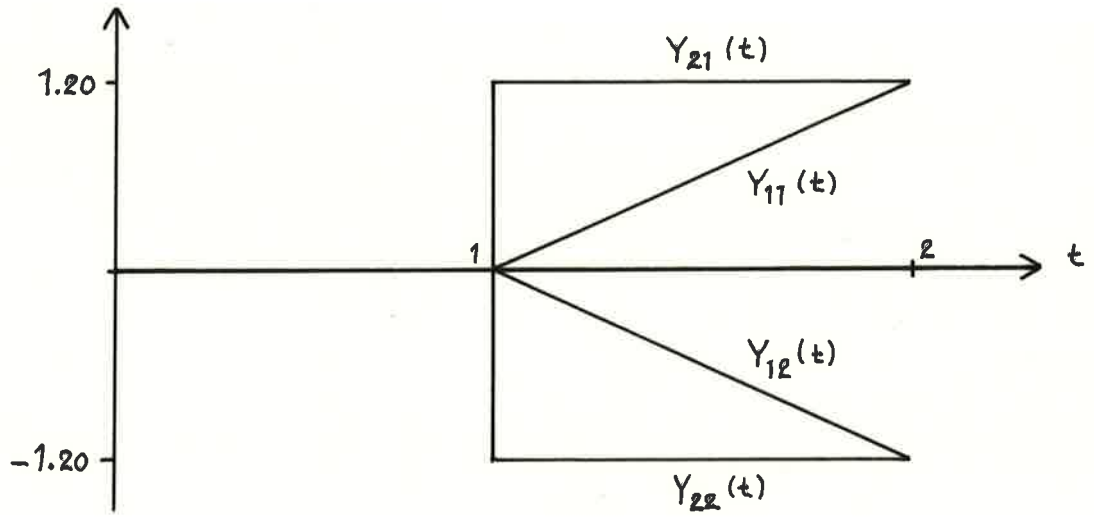
Figur 1 - Tillståndsvariabler



Figur 2 - Adjungerade variabler och styrsignal

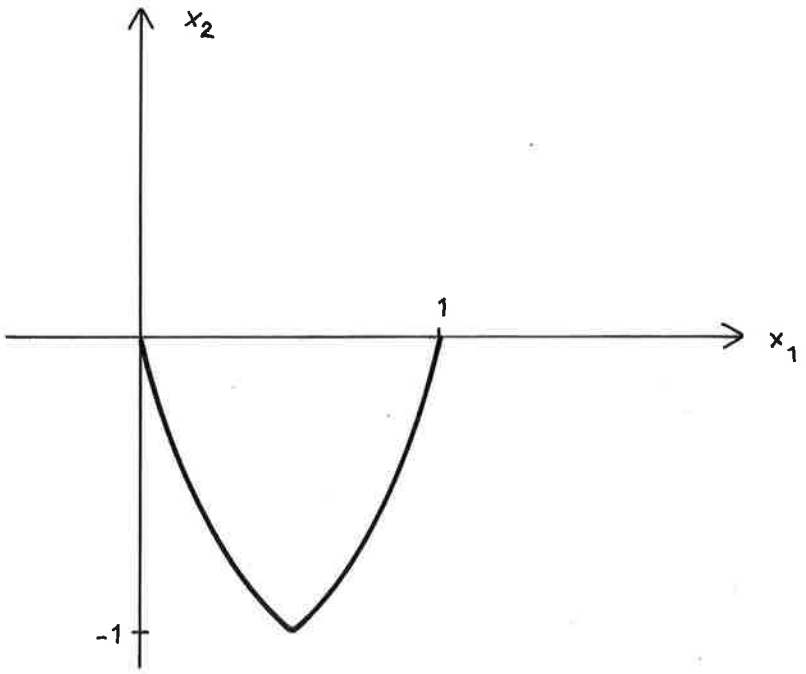


Figur 3 - Störningsvariabler Z



Figur 4 - Störningsvariabler Y

I fasplanet får optimaltrajektorian utseendet.



Figur 5 - Fasplan för dubbelintegratorn

Karakteristisk data för andra gissade värden på a_1 och a_2 framgår ur följande tabell

Gissat a_1	Gissat a_2	k	ε	antal iterationer	V	a_1	a_2
10.0	0.5	4.0	0.01	4	0.00002	11.05	11.05
2.0	1.0	0.1	0.01	3	0.001	1.63	1.65
1.5	0.5	0.1	0.005	4	0.002	1.55	1.57

Det är alltså uppenbart att a endast är bestämd till sin riktning, och kan godtyckligt multipliceras med någon positiv konstant. Förlustfunktionen V har alltså inget entydigt bestämt minimum, utan antar värdet noll om a har riktningen (1,1). Som tumregel för valet av k kan man säga att ju mer den gissade riktningen avviker från den optimala, desto större måste k vara för att få snabb konvergens.

b) Oscillator

Givet systemekvationerna

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -x_1 + u$$

sök en styrsignal så att systemet på minimaltid, $T = 5\pi$, överföres från (10,0) till origo. Adjungerade ekvationerna blir

$$\frac{dp_1}{dt} = p_2 \qquad p_1(0) = a_1$$

$$\frac{dp_2}{dt} = -p_1 \qquad p_2(0) = a_2$$

Hamiltonfunktionen

$$H = p_1x_2 - p_2x_1 + p_2u$$

Maximum fås då

$$u = \text{sign}\,(p_2)$$

Man har alltså

$$f_1 = x_2 \qquad g_1 = p_2$$

$$f_2 = -x_1 + \text{sign}\,(p_2) \qquad g_2 = -p_1$$

vilket ger störningsekvationerna

$$\frac{d}{dt} Y_{11} = Y_{21} \qquad Y_{11}(0) = 0$$

$$\frac{d}{dt} Y_{12} = Y_{22} \qquad Y_{12}(0) = 0$$

$$\frac{d}{dt} Y_{21} = -Y_{11} + (u(t) - u(t-h)) / (p(t) - p(t-h)) \cdot Z_{21} \qquad Y_{21}(0) = 0$$

$$\frac{d}{dt} Y_{22} = -Y_{12} + (u(t) - u(t-h)) / (p(t) - p(t-h)) \cdot Z_{22} \qquad Y_{22}(0) = 0$$

där $\frac{\partial f_2}{\partial p_2}$ har approximerats med $(u(t) - u(t-h)) / (p(t) - p(t-h))$.
h är som förut steglängden i Runge-Kutta-rutinen. För den andra uppsättningen störningsekvationer fås

$$\frac{d}{dt} Z_{11} = Z_{21} \qquad Z_{11}(0) = 1$$

$$\frac{d}{dt} Z_{12} = Z_{22} \qquad Z_{12}(0) = 0$$

$$\frac{d}{dt} Z_{21} = -Z_{11} \qquad Z_{21}(0) = 0$$

$$\frac{d}{dt} Z_{22} = -Z_{12} \qquad Z_{22}(0) = 1$$

Som förut väljes förlustfunktionen

$$V = x_1^2(T) + x_2^2(T)$$

och alltså

$$\frac{dV}{da} = \left(\frac{dV}{da_1}, \frac{dV}{da_2} \right) = \left(\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2} \right) \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

a_1 och a_2 skall alltså uppdateras enligt

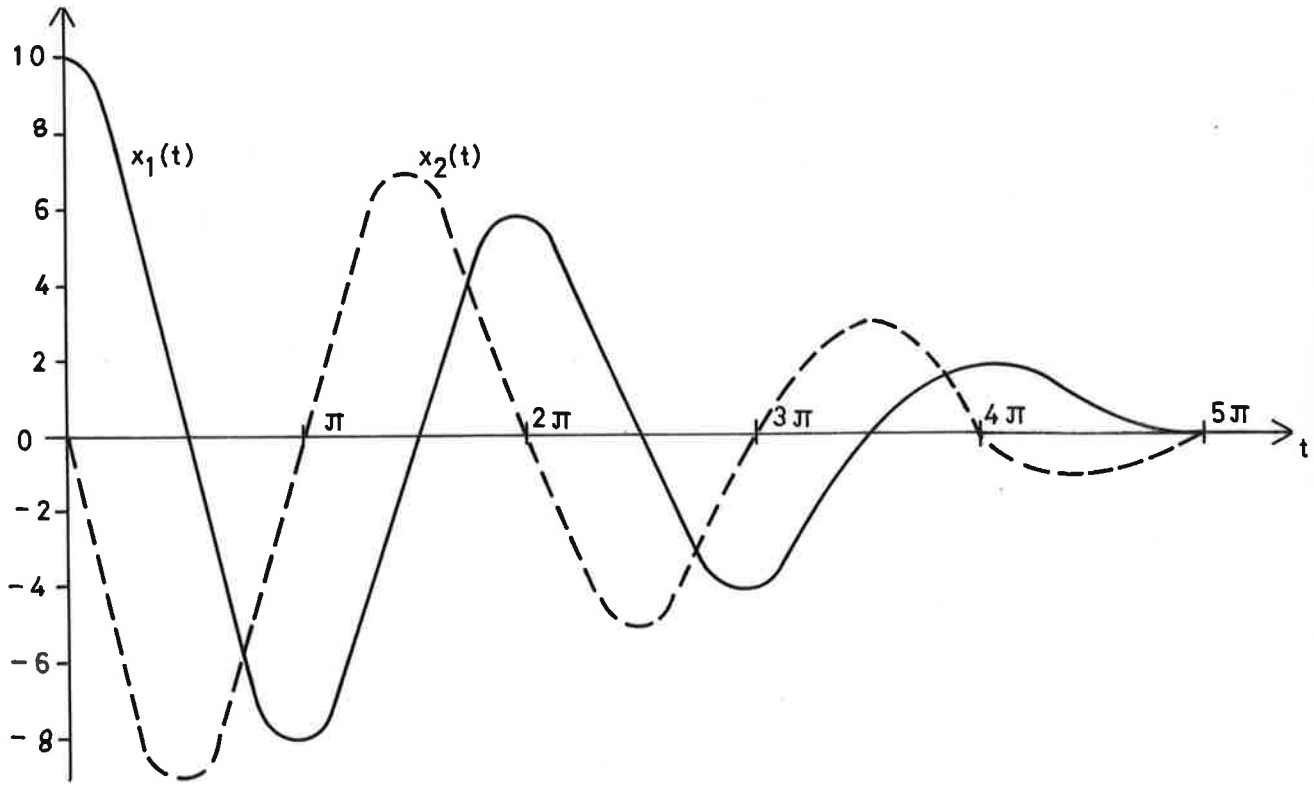
$$a_{1\ new} = a_{1\ old} - 2k \cdot (x_1(T) \cdot Y_{11}(T) + x_2(T) \cdot Y_{21}(T))$$

$$a_{2\ new} = a_{2\ old} - 2k \cdot (x_1(T) \cdot Y_{12}(T) + x_2(T) \cdot Y_{22}(T))$$

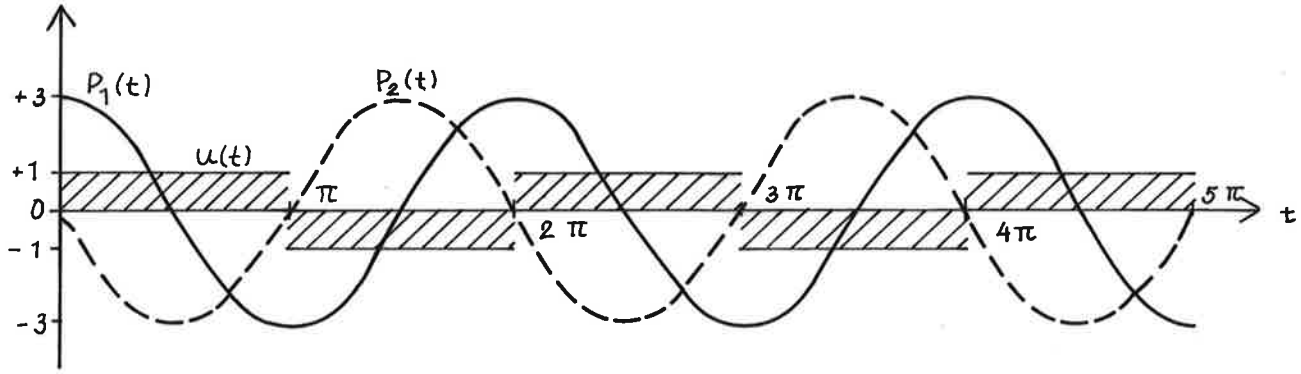
Matrisen A har i detta fall rent imaginära egenvärden, varför man inte kan säga något om hur många gånger styrsignalen kan byta tecken. Liksom för dubbelintegratorn har i programmet u valts = - sign(p_2), vilket ger minimum i stället för maximum åt Hamiltonfunktionen. Adjungerade variablerna har alltså även här motsatt tecken.

Med initialgissningar $a_1 = 2.0$, $a_2 = 1.0$ och $k = 0.1$ hade efter fyra iterationer erhållits $a_1 = 3.12$ och $a_2 = 0.06$ vilket gav $V = 0.04$.

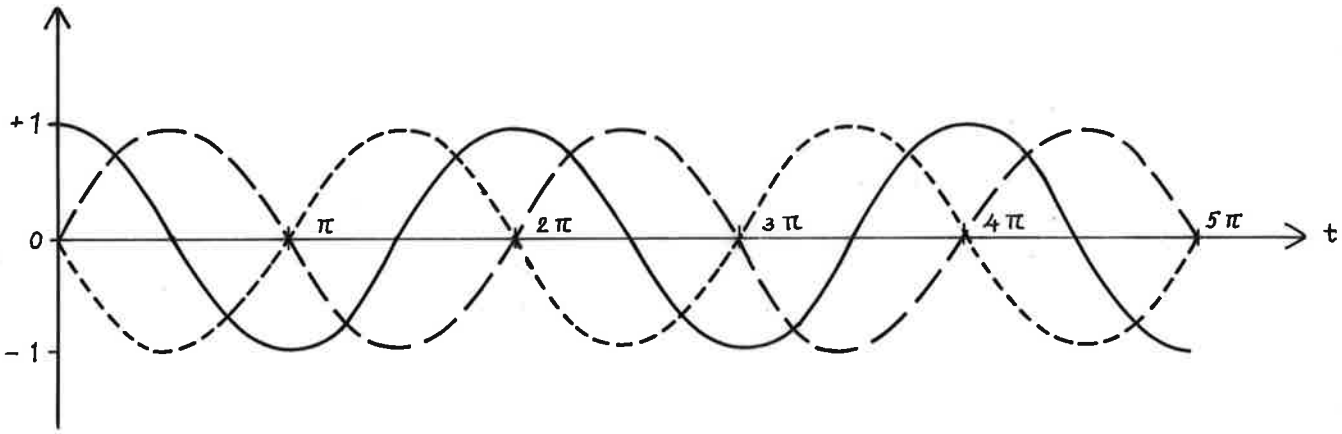
I fig. 1 visas tillståndsvariablerna som funktion av tiden, och i fig. 2 adjungerade variablerna med den optimala styrsignalen.



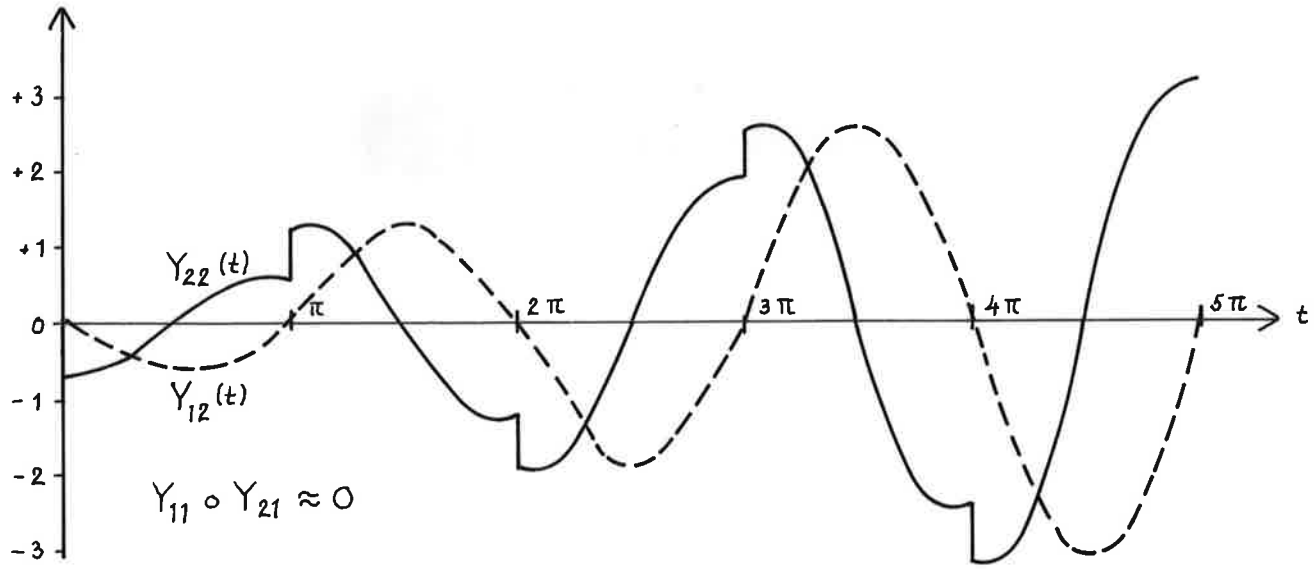
Figur 1 - Tillståndsvariabler $x_1(t)$ och $x_2(t)$



Figur 2 - Adjungerade variabler med optimal styrsignal



Figur 3 - Störningsvariabler Z_{11} och Z_{22} (heldragen), Z_{12} (långstreckad) samt Z_{21} (kortstreckad)



Figur 4 - Störningsvariabler Y

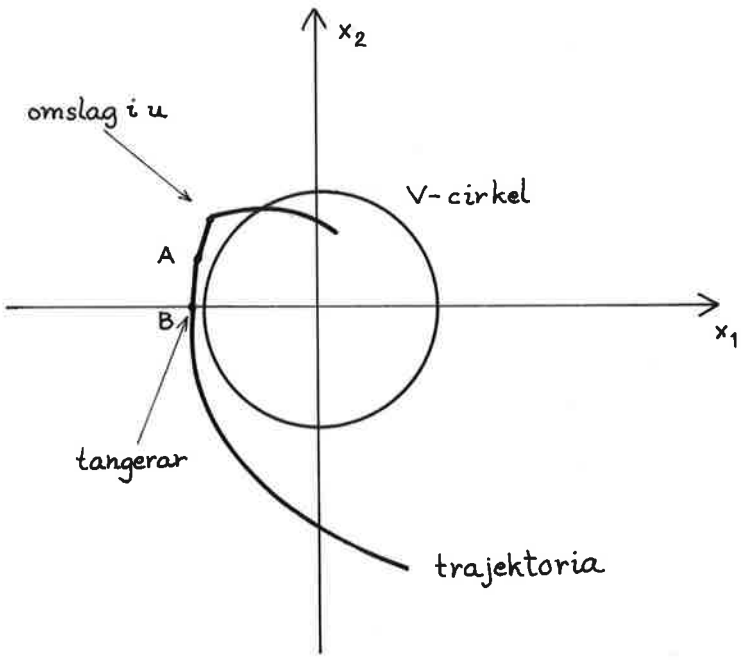
På samma sätt som i förra exemplet visar det sig att V inte har ett entydigt bestämt minimum, utan a är endast bestämd till sin riktning. Denna riktning sammanfaller med positiva a_2-axelns. Några konvergenssvårigheter uppstod inte för val av a som skiljde sig mycket från de optimala, utan efter 4-5 iterationer hade i regel önskad noggrannhet uppnåtts.

Med kännedom om optimaltrajektoriernas utseende är det nu möjligt att förklara varför metoden med ieration även på tiden inte fungerar. Betrakta exempelvis oscillatorn. Det gäller då

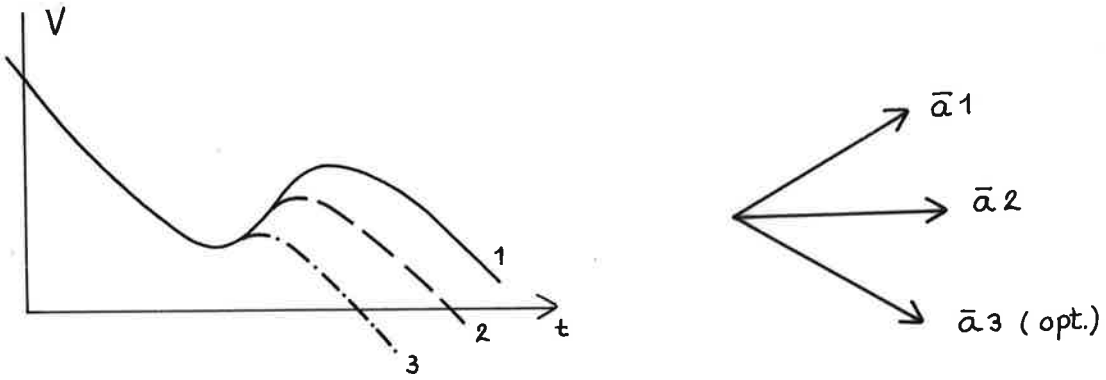
$$\frac{dV}{dt} = 2x_2 u$$

$\frac{dV}{dt}$ är alltså noll för x_2 lika med noll.

I det optimala fallet då u växlar tecken samtidigt som x_2 är tydligen de singulära punkterna terasspunkter med avseende på t vid fixa a_1 och a_2. Är emellertid styrningen blott nästan optimal, kan följande situation föreligga.



V har nu ett minimum med avseende på t. Att vid iterationen ha kommit till ett läge A innebär då att tiden tf minskas, samtidigt som ändringarna i a strävar att på denna tid göra förlustfunktionen så liten som möjligt. Det a som minimerar V är emellertid lika med det optimala värdet på a vilket inses av att $\frac{dV}{dt}$ i det optimala fallet alltid göres så negativ som möjligt. Omslagspunkten för u kommer alltså att röra sig mot B. V som funktion av t längs trajektorian kommer alltså att få följande utseende med ökat antal iterationer.

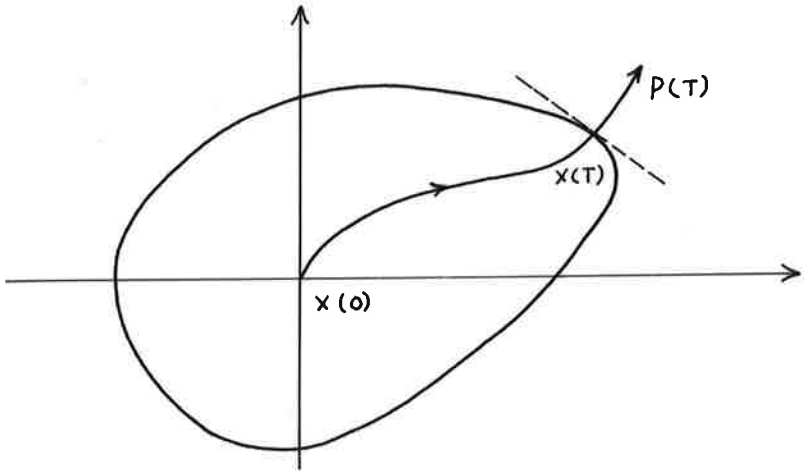


För att minimat skall omvandlas till terasspunkt och således möjliggöra en ökning av tf, krävs alltså att man nästan exakt uppsöker de värden på tf och a som ger minimat, vilket måste innebära en successiv minskning av konstanten k. När man väl förvandlat minimat till en punkt är alltså k så liten att den är verkningslös.

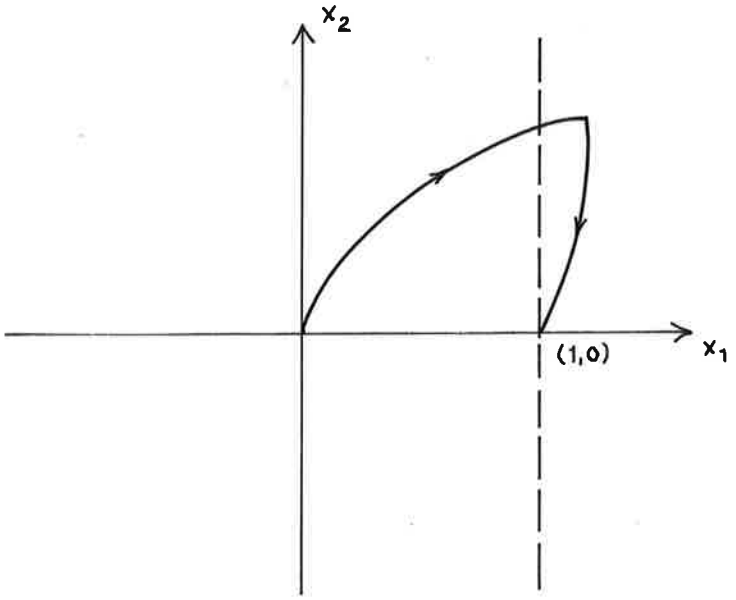
Med mycket goda initialgissningar på a och tf är det emellertid möjligt att få systemet att gå in mot en sådan här kritisk punkt som samtidigt är det sökta sluttillståndet. Det kräver emellertid att man praktiskt taget känner den sökta minimaltiden.

3.2 Algoritm för numerisk lösning av tidsoptimal styrning med icke-känt T.

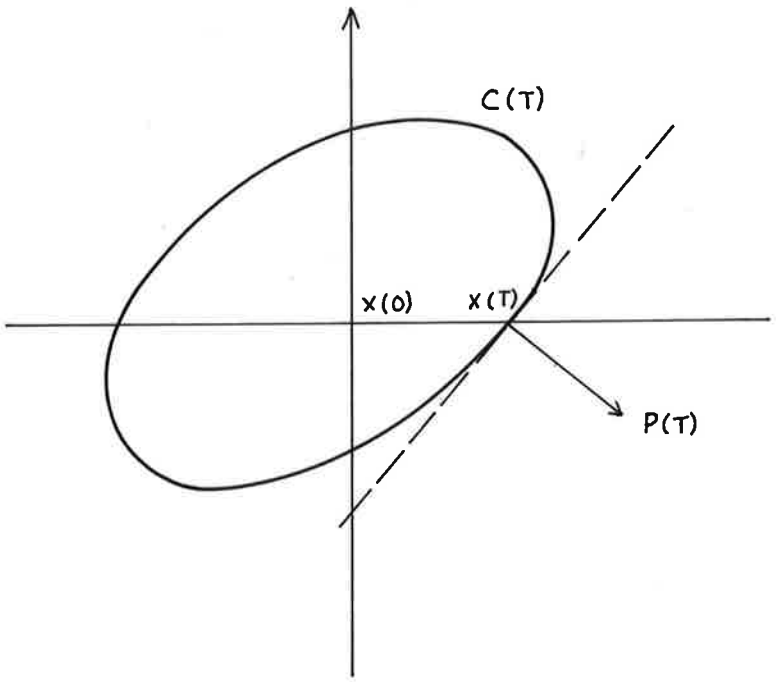
Betrakta åter den tidigare definierade mängden C(t). Det visades att C(t) är konvex och sluten, samt att randpunkterna utgjordes av punkter som nåtts genom optimal styrning. Vidare visades att adjungerade variabeln p(t) i varje ögonblick är den utåtriktade normalen till C(t). Har man funnit den optimala styrsignal som överför systemet till x(T) har man då följande



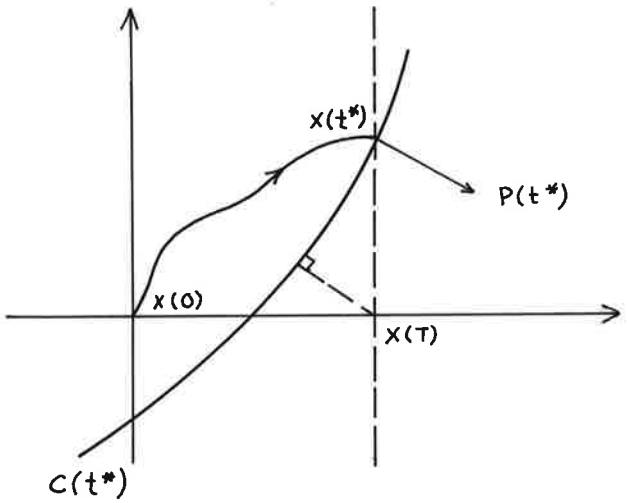
Grundidén som anknyter sig till föregående algoritm är nu följande. Antag att man i stället för att integrera tiden T integrerar systemet till dess att man träffar någon linje genom x(T) och sedan längs denna linje definierar en förlustfunktion som förut. Med störningsekvationerna skulle det då vara möjligt att avgöra hur begynnelsevärdet a skall ändras för att man vid nästa integration skall träffa denna linje närmare det sökta sluttillståndet x(T). Betrakta exempelvis dubbelintegratorn, och antag att det gäller att styra systemet från origo till punkten (1,0).



Ett möjligt val vore då linjen $x_1 = 1$, det vill säga integrera varje gång systemet till dess att $x_1 = 1$ och modifiera sedan a så att man nästa gång träffar linjen närmare (1,0). Vid ett sådant val har man emellertid inte garderat sig mot möjligheten att den sökta optimaltrajektorian vid någon tidpunkt är sådan att $x_1(t) > 1$. Är detta fallet är det uppenbart att resonemanget ovan inte fungerar. Ur konvexiteten hos C(t) följer emellertid att det existerar en linje sådan att optimaltrajektorian förlöper helt mellan x(0) och denna linje genom x(T). Denna linje är uppenbarligen tangenten till C(t) i x(T).



Genom att utnyttja konvexiteten hos C(t) kan man nu ange en algoritm som med utgångspunkt från en gissad initialtangent, exempelvis $x_1 = x_1(T)$, ger den sökta tangenten i x(T). Det kommer då att visa sig att man även löst problemet, det vill säga funnit det a som implicit överför systemet till x(T), utan att behöva tillgripa ytterligare iteration med hjälp av en förlustfunktion. Antag först tiden fixerad till t^*, och att följande situation råder



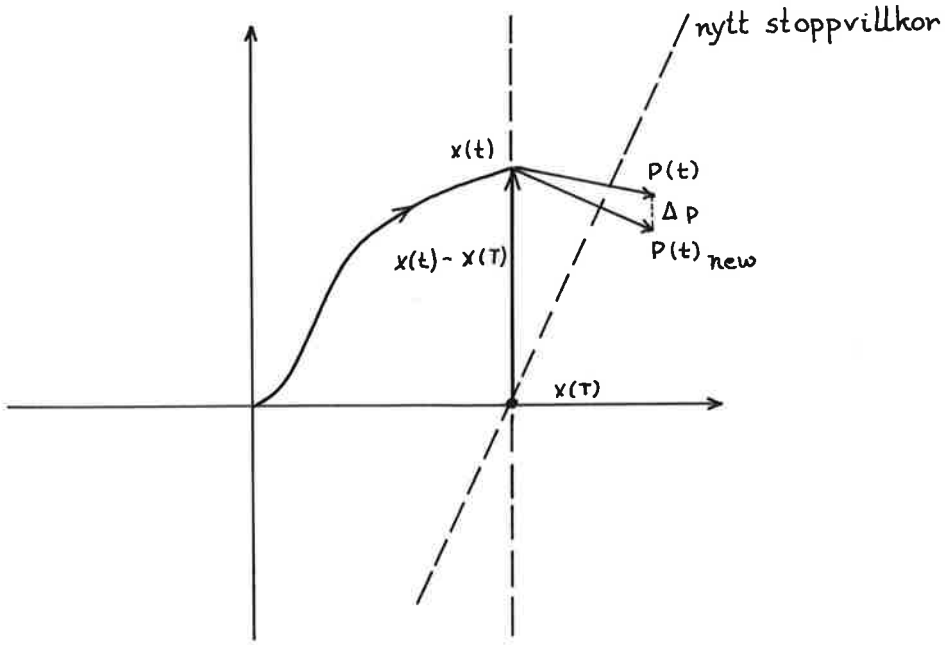
Om konturen $C(t^*)$ vet man att den är konvex, och det val av a som gör avståndet från $x(t^*)$ till x(T) så litet som möjligt, är uppenbarligen det som gör att $p(t^*)$ sammanfaller med $x(T) - x(t^*)$. För att få dessa riktningar att sammanfalla bör alltså $p(t^*)$ uppdateras enligt

$$p(t^*)_{new} = p(t^*)_{old} - k \cdot (x(t^*) - x(T)) \qquad (3.6)$$

Observera att $p(t^*)$ är intressant endast till sin riktning och inte till sin storlek. Släpp nu kravet att tiden är fixerad till t^*, och välj i stället att integrera systemet fram till ett första stoppvillkor $x_1(t) \geq x_1(T)$. Datera upp p(t) enligt (3.6) och sök med hjälp av störningselementen $\frac{dp_i}{da_j}$ det a som ger $p(t)_{new}$.

Om det skulle slumpa sig så väl att detta a är det sökta optimala, måste uppenbarligen stopplinjen ändras för att man skall kunna vara säker på att hela optimaltrajektorian ligger mellan x(0) och linjen. Den enda linje som garanterar detta är linjen vinkelrät mot $p(t)_{new}$ och detta blir alltså det nya stoppvillkoret. Detta kan skrivas

$$< (x(t) - x(T)) \mid (p(t)_{new}) \geq 0 \qquad (3.7)$$



Tiden t kommer nu inte att vara densamma, och det är inte självklart att metoden konvergerar mot det minsta avståndet till x(T), det vill säga just x(T). Neustadt {4} visar emellertid att om den konvergerar, ger den en riktning åt p(0) = a, sådan att motsvarande styrsignal given ur maximiprincipen, är den optimala. Vi sammanfattar ovanstående i följande algoritm .

1. Välj ett stoppvillkor, $x_i(t) \geq x_i(T)$, och ett a så att systemet träffar linjen.

2. Integrera systemekvationerna, adjungerade ekvationerna och störningsekvationerna tills stoppvillkoret uppfyllts. Normalen till C(t) ges då av $p_2(t)$.

3. Bilda en förlustfunktion $V = (x(t) - x(T))^2$. Om $V < \varepsilon$ är problemet löst.

4. p(t) ändras enligt $p_{new} = p(t)_{old} - k(x(t) - x(T)) =$

$$= p(t)_{old} - k \cdot \Delta p(t)$$

5. För att orsaka $\Delta p(t)$ skall a uppdateras. Det gäller

$\Delta p = Z \cdot \Delta a$

eller i komponentform (n = Z)

$$\begin{aligned} \Delta p_1 &= \frac{dp_1}{da_1} \cdot \Delta a_1 + \frac{dp_1}{da_2} \cdot \Delta a_2 = Z_{11} \cdot \Delta a_1 + Z_{12} \cdot \Delta a_2 \\ \Delta p_2 &= \frac{dp_2}{da_1} \cdot \Delta a_1 + \frac{dp_2}{da_2} \cdot \Delta a_2 = Z_{21} \cdot \Delta a_1 + Z_{22} \cdot \Delta a_2 \end{aligned} \qquad (3.7)$$

Lös ut Δa_1 och Δa_2

$$\Delta a_1 = \frac{Z_{22} \cdot \Delta p_1 - Z_{12} \cdot \Delta p_2}{Z_{11} \cdot Z_{22} - Z_{21} \cdot Z_{12}}$$

$$\Delta a_2 = \frac{Z_{11} \cdot \Delta p_2 - Z_{21} \cdot \Delta p_1}{Z_{11} \cdot Z_{22} - Z_{21} \cdot Z_{12}}$$

Detta kräver alltså det $Z \neq 0$, och i det allmänna fallet (n > 2) inversion av matrisen Z. p(0) = a modifieras sedan enligt

$a_{new} = a_{old} + \Delta a$

6. Nytt stopvillkor gives av

$$< (x(t) - x(T)) \mid p_{new} >$$

7. Återvänd till 2.

Som kommer att visas i testexemplen är ett nödvändigt krav att systemets begynnelsetillstånd $x(0)$ är sådant att $Ax(0) = 0$. Styrsignalen $u(t) \equiv 0$ orsakar då ingen ändring i systemet och man vet att $x(0)$ alltid tillhör $C(t)$, som då blir en mängd som kan sägas sprida sig runt $x(0)$.

Testexempel

a) Dubbelintegrator

Givet systemekvationerna

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = u$$

Sök styrsignalen $u(t)$ så att systemet på minimal tid överföres från $(1,0)$ till origo. Villkoret $A \cdot x(0) = 0$ är uppfyllt. Adjungerade ekvationerna fås som förut

$$\frac{dp_1}{dt} = 0 \qquad p_1(0) = a_1$$

$$\frac{dp_2}{dt} = -p_1 \qquad p_2(0) = a_2$$

Hamiltonfunktionen

$$H = p_1 x_1 + p_2 u$$

ger $u = \text{sign}\,(p_2)$

Störningselementen satisfierar

$$\frac{dZ_{11}}{dt} = 0 \qquad Z_{11}(0) = 1$$

$$\frac{dZ_{12}}{dt} = 0 \qquad Z_{12}(0) = 0$$

$$\frac{dZ_{21}}{dt} = - Z_{11} \qquad Z_{21}(0) = 0$$

$$\frac{dZ_{22}}{dt} = - Z_{12} \qquad Z_{22}(0) = 1$$

Det gäller

$$\frac{d}{dt} (Z_{11} Z_{22} - Z_{12} Z_{21}) =$$

$$= \frac{dZ_{11}}{dt} \cdot Z_{22} + Z_{11} \cdot \frac{dZ_{22}}{dt} - \frac{dZ_{12}}{dt} \cdot Z_{21} - Z_{12} \cdot \frac{dZ_{21}}{dt} = 0$$

Med $Z(0) = I$ följer då $\det(Z) = 1$ för alla t.

Förlustfunktionen V som nu endast har till uppgift att tala om hur nära det önskade sluttillståndet systemet befinner sig väljes

$$V = x_1^2(t) + x_2^2(t)$$

Programmet återfinnes i appendix 2.

Konvergensen för olika initialgissningar på a och för olika K redovisas i fig. 1-3.

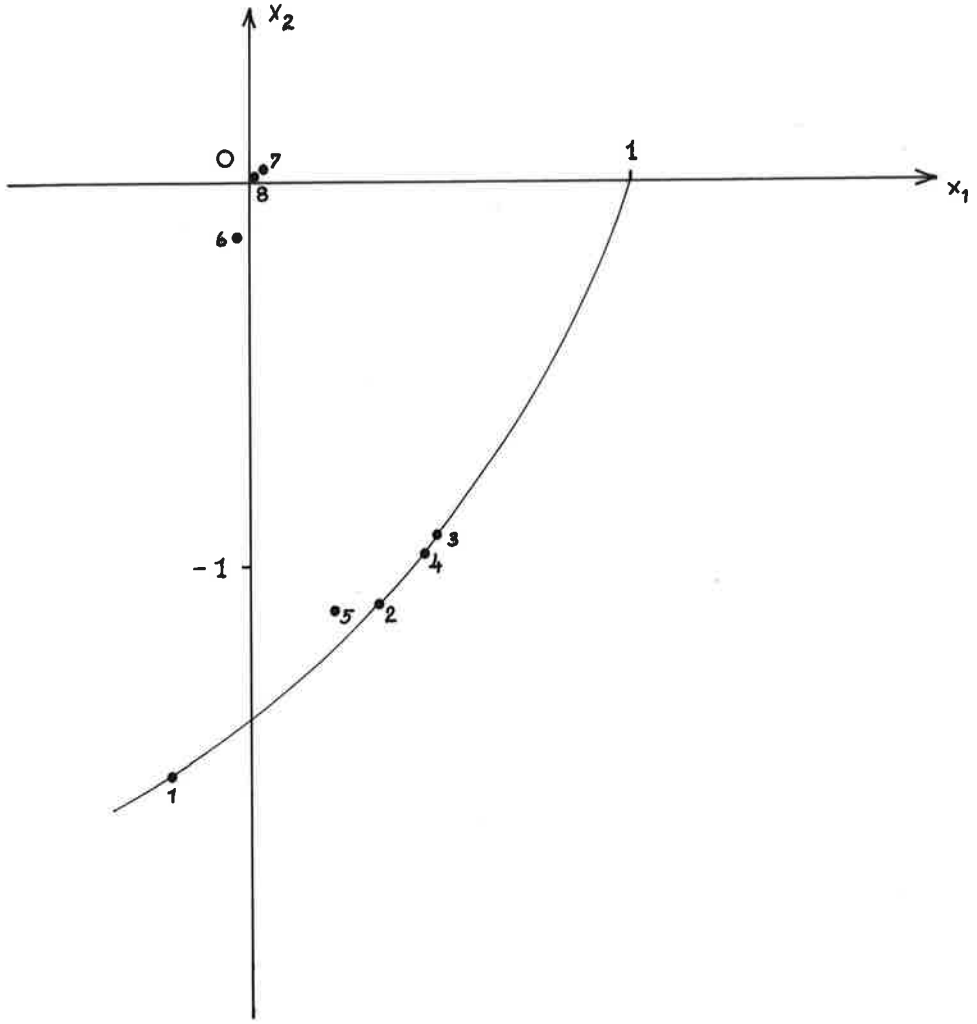


Fig. 1

a initial = (-1, -2), K = 0.1, ε = 0.001, h = 0.1

Iterationerna 1-4 ligger på den parabel i fasplanet som svarar mot u = -1. Detta beror på att initialgissningen är sådan att det behövs en ganska stor ändring av riktningen för att u skall byta tecken. Detta sker inte förrän efter tredje iterationen. Den optimala riktningen på a blev (-0.71, -0.70), och den sökta minimaltiden som väntat approximativt T = 2.0 (2.005). En begränsning i noggrannheten är naturligtvis den relativt stora integrationslängden. Detta är emellertid tyvärr nödvändigt på grund av långsamheten hos SMIL.

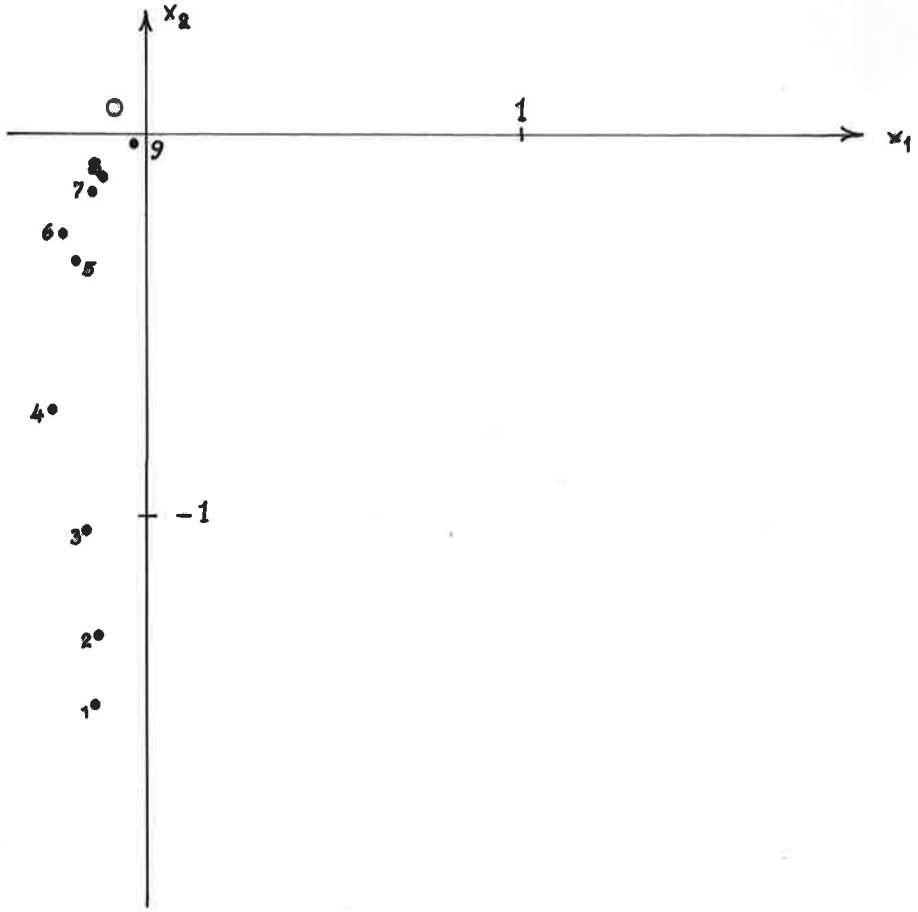


Fig. 2

$a_{initial}$ = (-1.0, 1.5), K = 0.05, ε 0.01, h = 0.1

$a_{initial}$ har här valts så att man undviker återgången längs parabelbågen, och iterationerna för stadigt systemet mot origo.

En minskning av K i fig. 1 medför iteration enligt fig. 3.

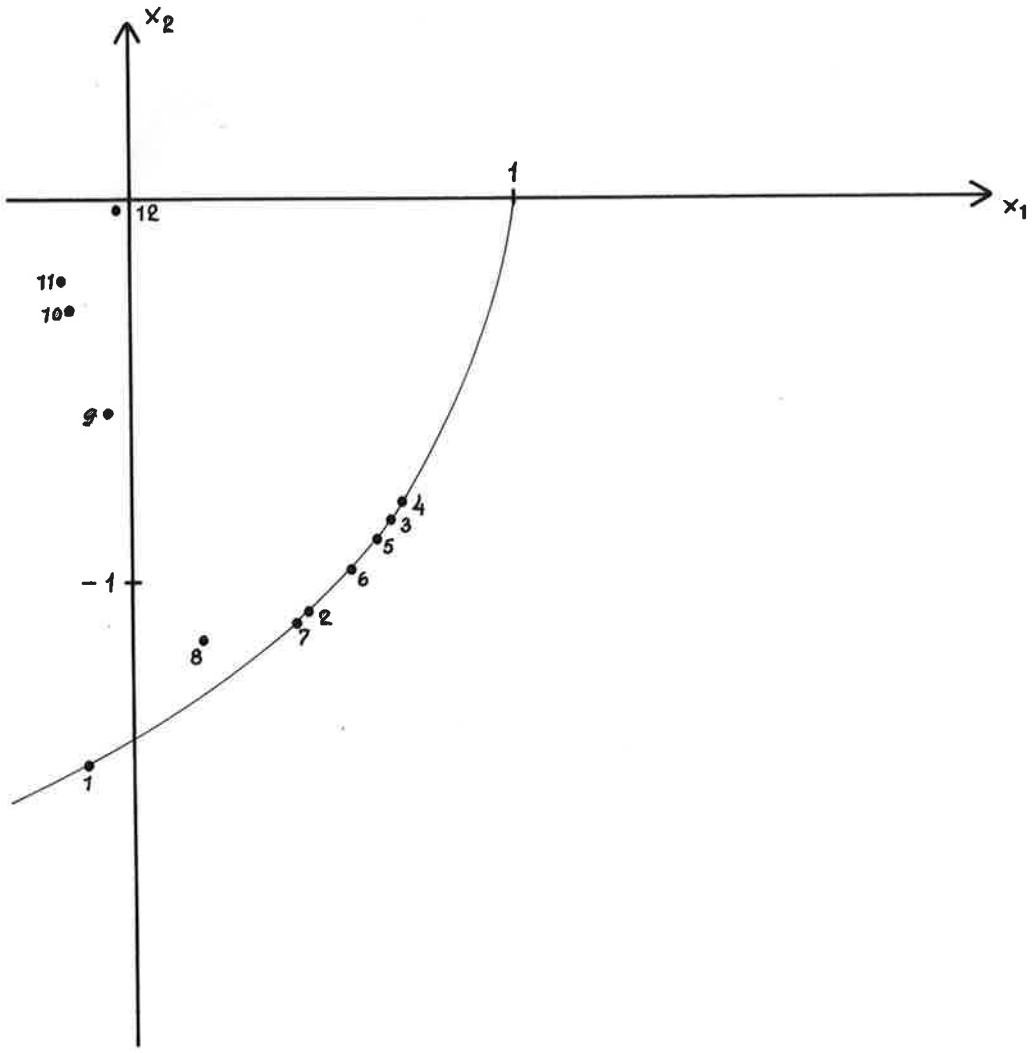


Fig. 3

$a_{initial}$ = (-1.0, -2.0), K = 0.05, ε 0.01, h = 0.1

Det krävs alltså betydligt fler iterationer för att få en sådan riktning på a att u byter tecken. En C(2) - kontur till punkten (1,0) beräknades och visas i figur 4. Denna förklarar varför systemet närmar sig origo från tredje kvadranten.

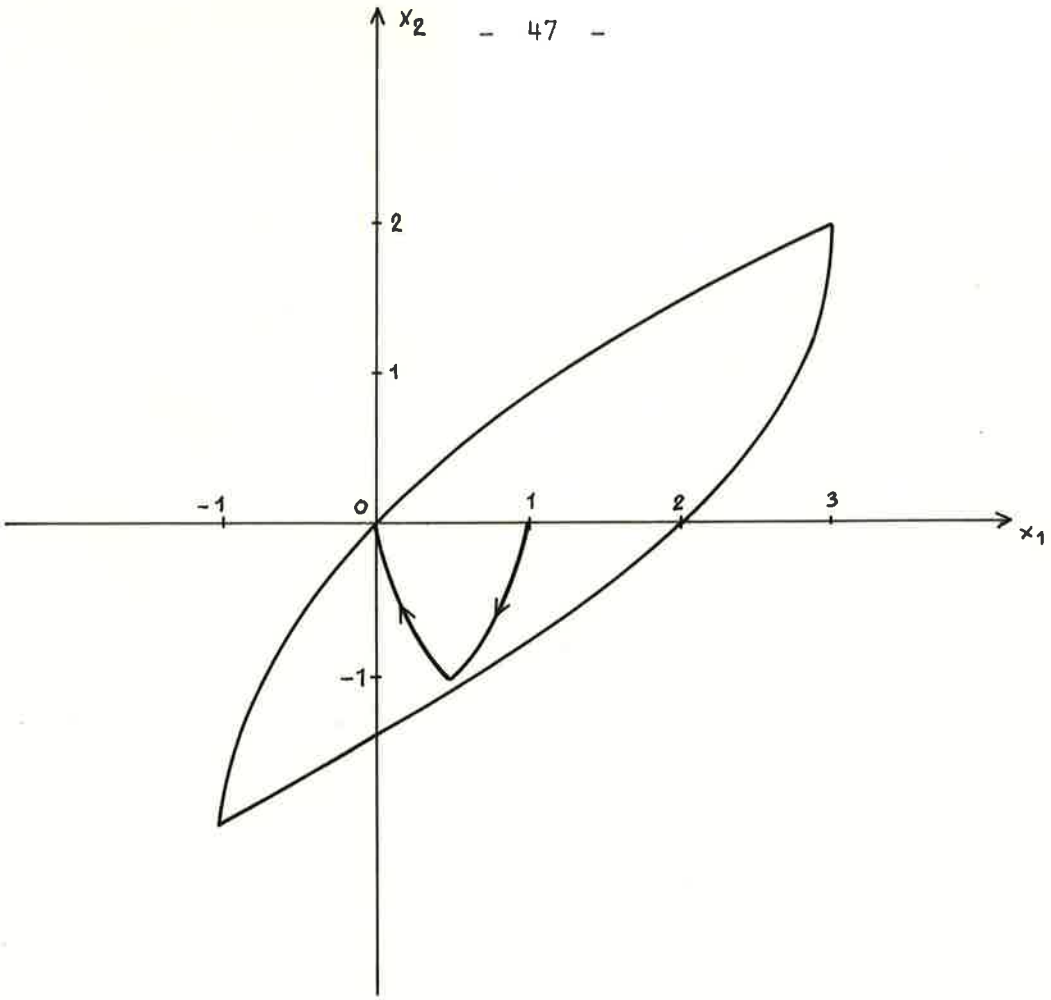


Fig. 4

C(2) - kontur till punkten (1,0) och optimaltrajektoria.

Kravet att $A \cdot x(0) = 0$ antyder att metoden bör vara mera lämpad för att beräkna optimal styrning för system med begynnelsetillstånd (0,0). Visserligen är $A \cdot x(0) = 0$ uppenbarligen ett alltför starkt krav, det är tillräckligt att det finns ett admissibelt u så att $A \cdot x(0) + Bu = 0$, men de möjliga begynnelsetillstånden är likväl begränsade. Detta är skillnaden mellan Neustadts {4} metod, och den ovan redovisade. Neustadt är begränsad till att sluttillståndet måste vara origo, medan det här krävs att begynnelsetillståndet måste vara sådant att $A \cdot x(0) + Bu = 0$ där $u \in U$, och under förutsättning att U innehåller origo är origo alltid ett möjligt sluttillstånd.

I fig. 5 visas iterationer vid styrning från origo till (1.0) och i fig. 6 C(2) - kontur till origo med optimaltrajektoria. Den optimala riktningen på a blev a = (0.69, 0.71) och tiden T = 2.0.

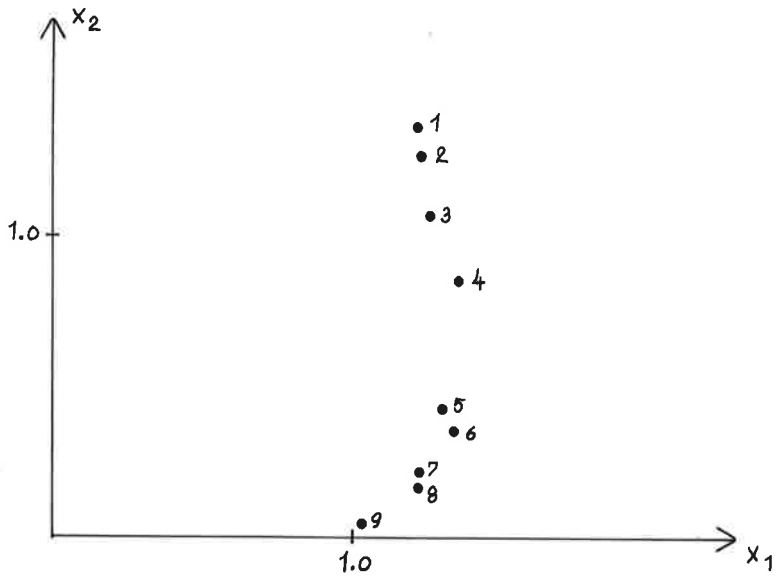


Fig. 5

$a_{initial}$ = (1.0, 1.5), K = 0.05, ε = 0.01, h = 0.1

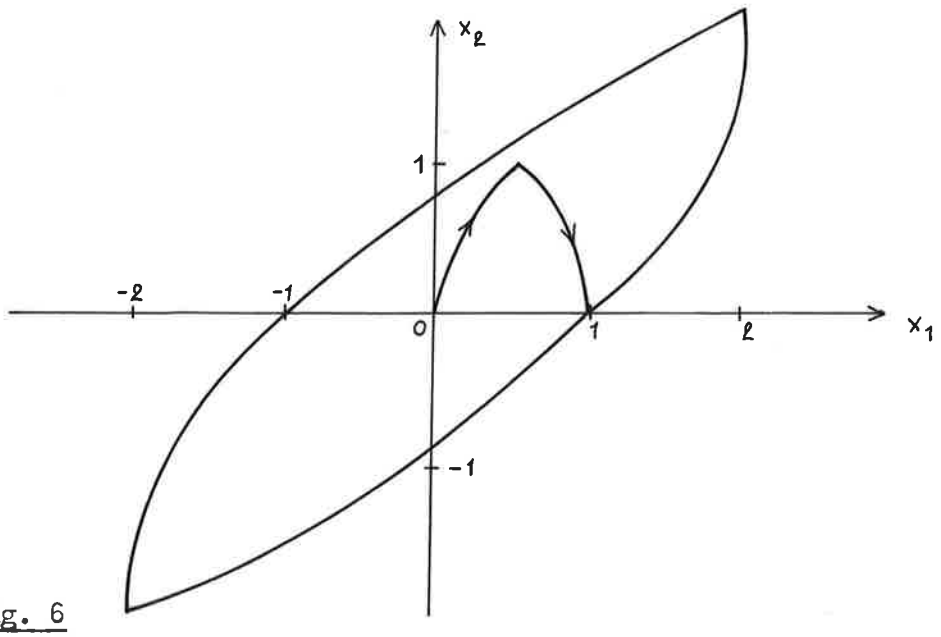


Fig. 6

C(2) - kontur till (0,0) och optimaltrajektoria.

För att studera inverkan av krökningen hos C(t) - konturen söktes optimala styrningen från origo till punkten (1,1). C(t) - kurvorna kan analytiskt visas vara parabler, och iterationerna bör då ligga på en skarpare krökt del av C(t) än vid styrning till (1,0). Detta visade sig också vara riktigt såtillvida att man med samma k(0.05) behövde många iterationer för att nå fram.Med en fördubbling av k till 0.1 nåddes emellertid sluttillståndet efter 8 iterationer, och med k = 0.2 redan efter 4 iterationer. Man kan alltså inte sägas ha några större svårigheter att välja k vid så relativt gynnsamma C(t) - ytor som det här är fråga om. Resultaten visas i fig. 7 och 8. Tiden bestämdes till T = 1.45 och a = (0.63, 0.77)

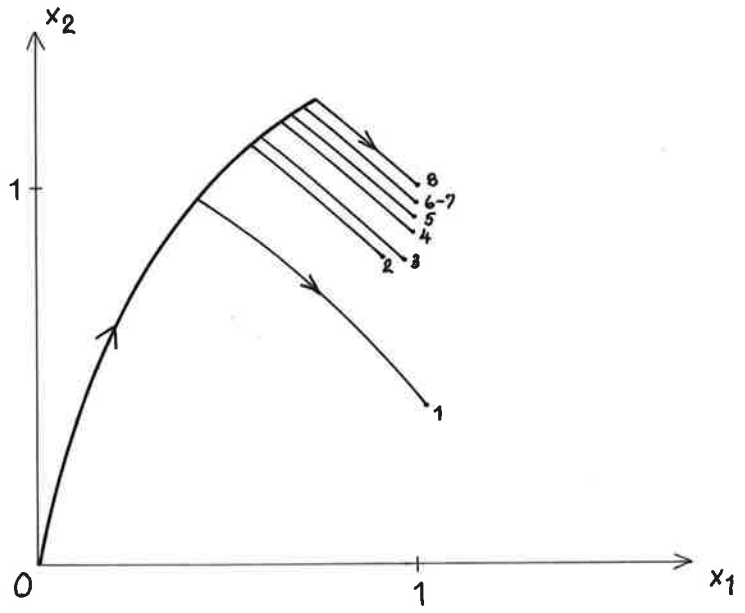


Fig. 7

$a_{initial}$ = (1.0, 1.05), k = 0.1, ε = 0.001, h = 0.04

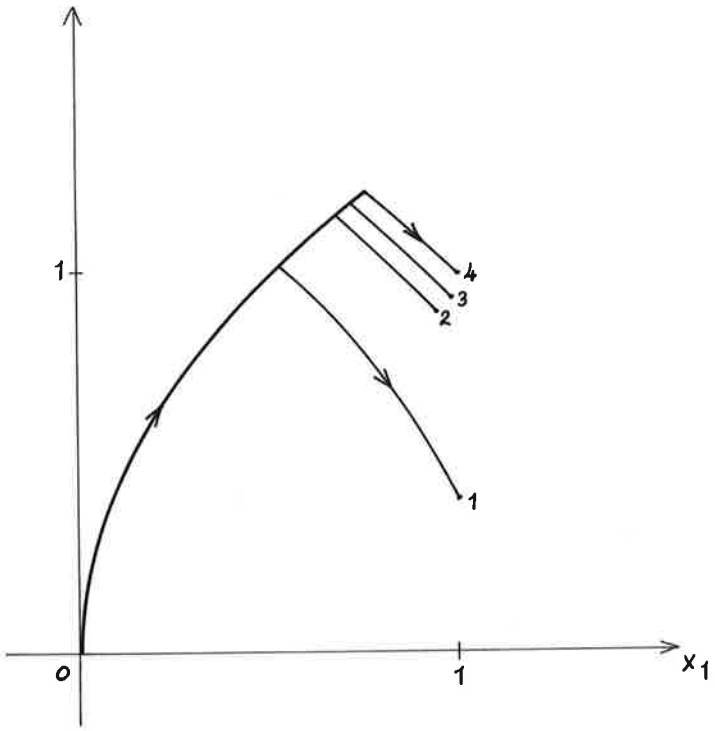


Fig. 8

$a_{initial}$ = (1.0, 1.05), K = 0.2, ε = 0.001, h = 0.04

b) Oscillator

Givet systemekvationerna

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -x_1 + u$$

Sök styrsignal u(t) så att systemet på minimaltid överföres från origo till (6.0). Adjungerade ekvationerna

$$\frac{dp_1}{dt} = p_2 \qquad p_1(0) = a_1$$

$$\frac{dp_2}{dt} = -p_1 \qquad p_2(0) = a_2$$

Hamiltonfunktionen

$$H = p_1x_2 - p_2x_1 + p_2u$$

ger

$$u = \text{sign}\,(p_2)$$

Störningsekvationerna

$$\frac{dZ_{11}}{dt} = Z_{21} \qquad Z_{11}(0) = 1$$

$$\frac{dZ_{12}}{dt} = Z_{22} \qquad Z_{12}(0) = 0$$

$$\frac{dZ_{21}}{dt} = -Z_{11} \qquad Z_{21}(0) = 0$$

$$\frac{dZ_{22}}{dt} = -Z_{12} \qquad Z_{22}(0) = 1$$

Förlustfunktionen

$$V = (x_1(t) - 6)^2 + x_2(t)^2$$

Gissad initialriktning var a = (1.0, 0.5). K valdes lika med 0.2 och problemet ansågs löst då V < 0.01. Konvergensen framgår ur fig. 1.

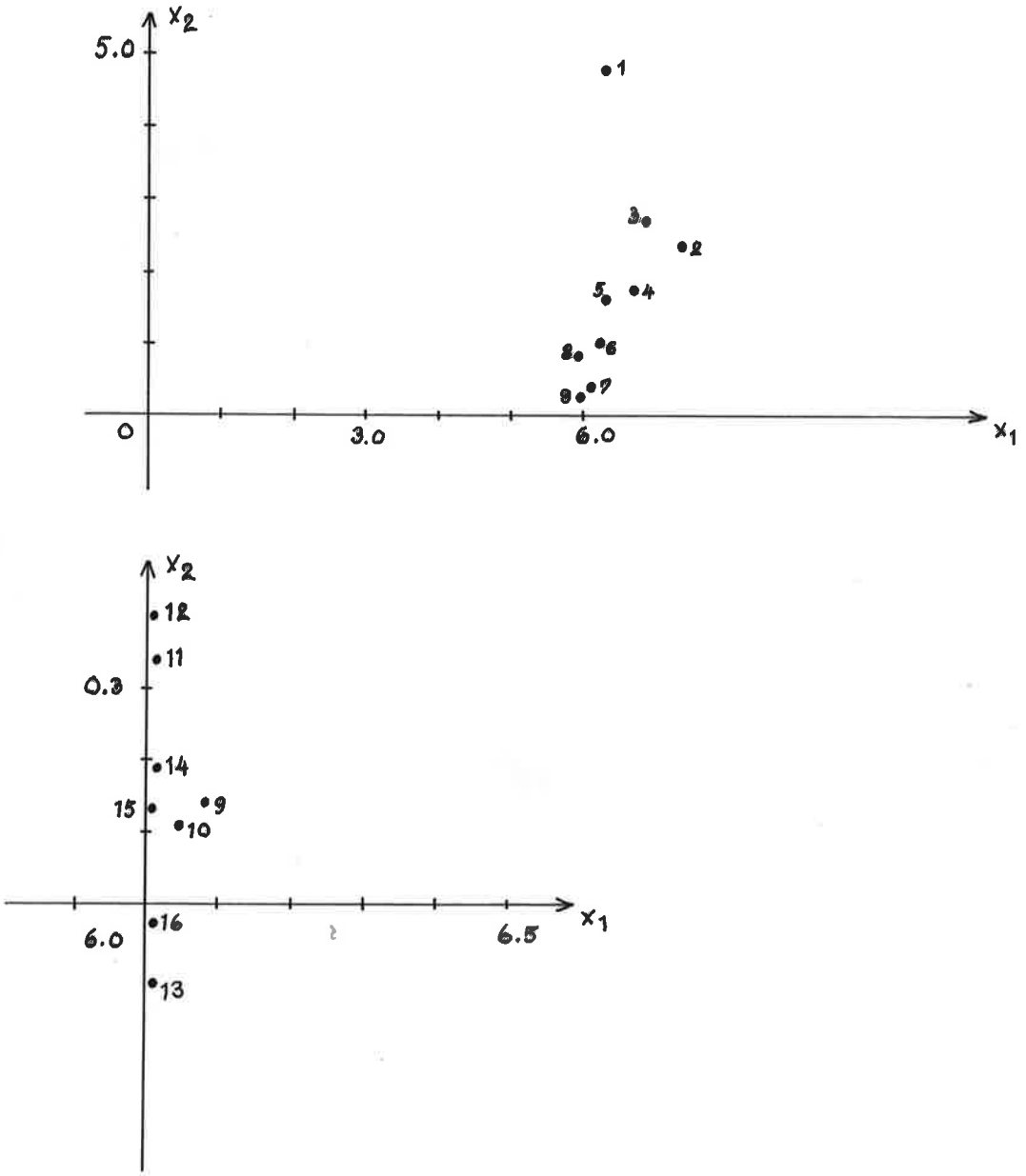


Fig. 1

Konvergens mot (6.0), h = 0.2

Efter 16 steg var V = 0.003 och a hade modifierats till a = (-0.99, -0.09), vilket är praktiskt taget motriktat den gissade initialriktningen. En god initialgissning är alltså inte nödvändigt, vilket ju också följer av det faktum att adjungerade ekvationerna är periodiska och att sålunda systemet alltid kommer att nå den första stopplinjen $x_1 \geq 6$. T bestämdes till 9.52 ($\approx 3\pi$). Optimaltrajektoria och C(3π)-kontur framgår ur fig. 2.

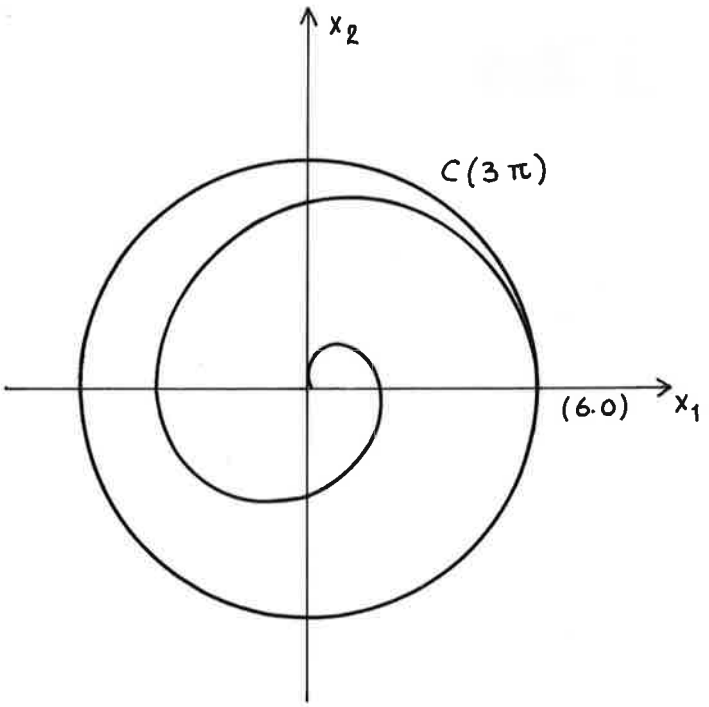


Fig. 2

Optimaltrajektoria och C(3π)-kontur.

C(t)-konturen blir för alla t en cirkel, och detta får följder för valet av K. Antag exempelvis att man vid en första integration når stopplinjen långt från punkten (6.0). p(t) kan då vara nära sammanfallande med linjen, och denna kommer då att ändras till nästa iteration på ett sådant sätt att man ständigt minskar tiden (se fig. 3) om K väljes för liten.

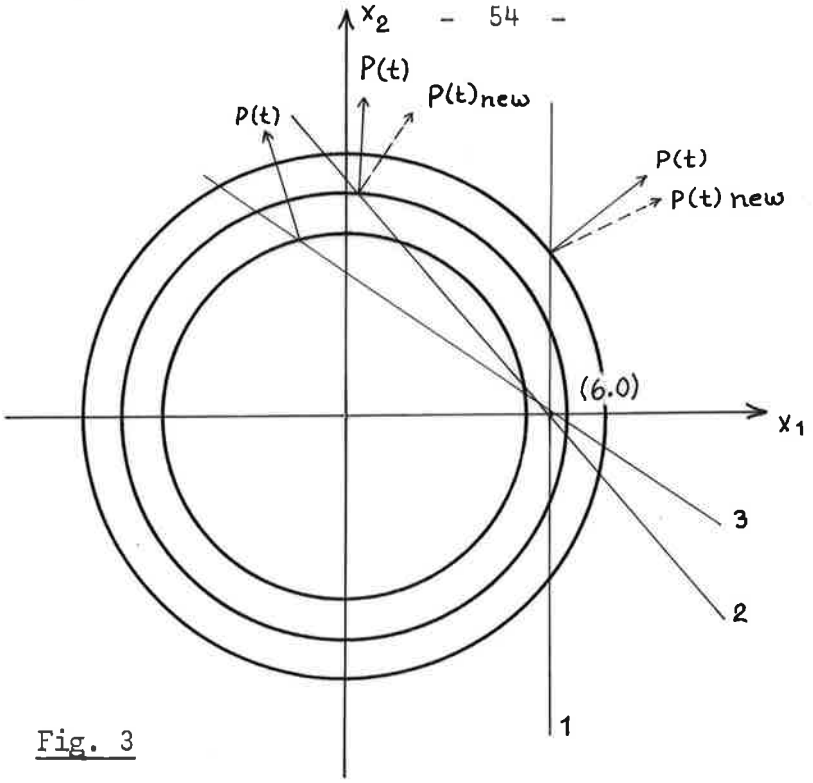


Fig. 3

Väljes å andra sidan K för stor kommer stopplinjen att oscillera kring den sökta tangentriktningen. Man är alltså mera begränsad i sitt val av K för oscillatorn än för dubbelintegratorn.

En annan uppenbar svårighet är att optimaltrajektorian tangerar C(3π)-konturen i (6.0). Eftersom man vid den numeriska lösningen är hänvisad till någorlunda stora integrationslängder, kan det för en nästan optimal lösning inträffa att då man befinner sig nära sluttillståndet, men dock icke så att stoppvillkoret uppfyllts, under nästa integration passerar stopplinjen och kommer tillbaka till den sida där $\langle (x(t) - x(T)) | p(t) \rangle < 0$. Detta skedde också då förlustfunktionen försöktes göras så liten som 0.0001 och med en steglängd h = 0.1. Detta bör naturligtvis kunna lösas genom att ytterligare minska integrationsintervallets längd men detta skulle innebära orimligt långa körtider på SMIL.

4. REFERENSER

{1} Pontryagin, L.S., et al. "The mathematical theory of optimal processes", New York 1962

{2} Rozonoer, L.I. "The maximum principle of L.S. Pontryagin in the theory of optimal systems", Automation and Remote Control 1959, vol. 20, 10, 11, 12.

{3} Levine, M.D. "A steepest descent method for synthesizing optimal control programmes", Proc. inst. of Mech. Engrs. 179 (1964-65) Port 3H (Nottingham Conference April 1965)

{4} Neustadt, L.W., Paiewonsky, B.H. "On synthesizing optimal controls", In Broida, V. editor "Automatic and Remote Control", Butterworths, London 1964

```
begin  integer ntot,np,p,antit,i,m,q,r;
real tin,te,t,tf,hin,h,V,Vold,eps,delta;
array yin,ye,y,z[1:25],Va,ka,a,aold[1:5];
;
```

```
procedure RK1ST(t,y,h,te,ye,p); value t; integer p; real t,te,h;
array y,ye;
begin integer j,k,n; real p2old,u,uold,sigma; array z,w[1:25],a[1:5];
procedure Fkt(x,y,z,konst); value x,konst; real x,konst; array y,z;
  begin comment systemekvationerna z[1] och z[2], adjungerade ekvationerna
  z[3] och z[4];
  z[1]:=y[2];
  z[2]:=-sign(y[4]);
  z[3]:=0;
  z[4]:=-y[3];
comment störningsekvationerna Y11,Y12,Y21,Y22;
  z[5]:=y[7];
  z[6]:=y[8];
  z[7]:='konst×y[11];
  z[8]:='konst×y[12];
comment störningsekvationerna Z11,Z12,Z21,Z22;
  z[9]:=0;
  z[10]:=0;
  z[11]:=-y[9];
  z[12]:=-y[10];
  end FKT;
comment nu börjar proceduren RK1ST;
sigma:=0; n:=0;
B: a[1]:=a[2]:=a[5]:=h/2; a[3]:=a[4]:=h; te:=t;
for k:=1 step 1 until ntot do ye[k]:=w[k]:=y[k];
p2old:=y[4]; uold:=-sign(p2old);
for j:=1 step 1 until 4 do
 begin
Fkt(te,w,z,sigma); te:=t+a[j];
  for k:=1 step 1 until ntot do
    begin
    w[k]:=y[k]+a[j]×z[k];
    ye[k]:=ye[k]+a[j+1]×z[k]/3
    end k
 end j;
u:=-sign(ye[4]);
if abs(u-uold)>0.5^r=0 then begin r:=1; h:=hin/m; go to B end;
if abs(u-uold)>0.5^r=1 then
  begin
    if n=0 then
      begin
        if ye[4]-p2old=0 then sigma:=0 else sigma:=(u-uold)/(ye[4]-p2old);
        n:=n+1; go to B
      end
      else
      begin
        n:=0; r:=0; h:=hin; go to C
      end
  end;
C:    if p=1 then
 begin
print(2,4,te); punch(8); print(2,2,u); punch(8);
for j:=1 step 1 until ntot do
  begin print(2,2,ye[j]); punch(0) end j;
punch(1);
end p
end RK1ST;
;
```

```
procedure FÖRLUST(V,n,A); value n; integer n; real V; array A;
begin integer i;
   V:=0;   for i:=1 step 1 until n do V:=V+A[i]×A[i]
end FÖRLUST;

procedure GRAD(V,Vold,Va,ka,y,n,B); value V,n; integer n;
real V,Vold; array Va,ka,y; label B;
begin integer i;
  if n=1 then go to A;
  if V<Vold then go to A;
  for i:=1 step 1 until np do ka[i]:=ka[i]/2;
  go to B;
A:  Vold:=V;
  Va[1]:=2×((y[1]×y[5])+(y[2]×y[7]));
  Va[2]:=2×((y[1]×y[6])+(y[2]×y[8]));
end procedure GRAD;

procedure NEWat(Va,ka,a,n); value n; integer n;
array Va,ka,a;
begin integer i;
   for i:=1 step 1 until n do a[i]:=a[i]-ka[i]×Va[i];
end NEWat;

procedure SKRIV(a,n1,b,n2,c,n3,d,e,f,g); value n1,n2,n3,d,e,f,g;
integer n1,n2,n3,f,g; real d,e; array a,b,c;
comment proceduren skriver ut de n1 första komponenterna i vektorn a, de n2
första i vektorn b, de n3 första i vektorn c. Vidare skrives de reella talen
d och e ut. Allt med f heltal och g decimaler;
begin integer i;
  for i:=1 step 1 until n1 do
    begin print(f,g,a[i]); punch(0)
    end;
  punch(8);
  for i:=1 step 1 until n2 do
    begin print(f,g,b[i]); punch(0)
    end;
  punch(8);
  for i:=1 step 1 until n3 do
    begin print(f,g,c[i]); punch(0)
    end;
  punch(8);
  print(f,g,d); punch(8); print(f,g,e); punch(1)
end SKRIV;
;
```

```
comment nu börjar själva programmet;

ntot:=read; np:=read; tin:=read; hin:=read; eps:=read; delta:=read;
p:=0; antit:=1; q:=1;r:=0;
for i:=1 step 1 until ntot do yin[i]:=read;
for i:=1 step 1 until np do ka[i]:=read;
tf:=read; m:=read;
a[1]:=yin[3]; a[2]:=yin[4];
INITIAL:
print(3,0,antit); punch(1); print(1,5,hin); punch(8);
print(2,5,a[1]); punch(0); print(2,5,a[2]);punch(1);
h:=hin; t:=tin;
for i:=1 step 1 until ntot do y[i]:=yin[i]; y[3]:=a[1]; y[4]:=a[2];
if p=1 then begin RK1ST(t,y,0,te,ye,p) end;
A1:
RK1ST(t,y,h,te,ye,p);
if (tf-te)<h then h:=tf-te;
if te≥tf then go to A2;
t:=te; for i:=1 step 1 until ntot do y[i]:=ye[i]; go to A1;
A2:
if p=1 then go to OPTIMAL;
FÖRLUST(V,np,ye);
print(2,5,V); punch(1);
if(q=1)^(V<delta) then begin q:=q+1; hin:=hin/2 end;
if V<eps then begin p:=1; punch(1); punch(1); go to INITIAL end;
GRAD(V,Vold,Va,ka,ye,antit,A3);
for i:=1 step 1 until np do aold[i]:=a[i];
go to A4;
A3:
for i:=1 step 1 until np do a[i]:=aold[i];
A4:
NEWat(Va,ka,a,np);
SKRIV(ye,4,a,2,ka,2,0,0,2,2);
antit:=antit+1; go to INITIAL;
OPTIMAL:
end
```

```
begin integer ntot,np,m,antit,q,p,i,b,r,s,m1;
real t,te,tin,h,hin,eps,delta,K,c1,c2,d,V,det;
array y,ye,yin,z[1:25],a,norm,slut,da,dp[1:5];
;
```

```
procedure RK1ST(t,y,h,te,ye,p); value t; integer p; real t,te,h;
array y,ye;
begin integer j,k,n; real p2old,u,uold,sigma; array z,w[1:25],a[1:5];
procedure Fkt(x,y,z,konst); value x,konst; real x,konst; array y,z;
  begin comment systemekvationerna z[1] och z[2], adjungerade ekvationerna
  z[3] och z[4];
  z[1]:=y[2];
  z[2]:=sign(y[4]);
  z[3]:=0;
  z[4]:=-y[3];
comment störningsekvationerna Z11,Z12,Z21,Z22;
  z[5]:=0;
  z[6]:=0;
  z[7]:=-y[5];
  z[8]:=-y[6];
  end FKT;
comment nu börjar proceduren RK1ST;
sigma:=0; n:=0;
B: a[1]:=a[2]:=a[5]:=h/2; a[3]:=a[4]:=h; te:=t;
for k:=1 step 1 until ntot do ye[k]:=w[k]:=y[k];
p2old:=y[4]; uold:=-sign(p2old);
for j:=1 step 1 until 4 do
 begin
Fkt(te,w,z,sigma); te:=t+a[j];
  for k:=1 step 1 until ntot do
    begin
    w[k]:=y[k]+a[j]×z[k];
    ye[k]:=ye[k]+a[j+1]×z[k]/3
    end k
 end j;
u:=-sign(ye[4]);
if abs(u-uold)>0.5^r=0 then begin r:=1; h:=hin/m; go to B end;
if abs(u-uold)>0.5^r=1 then
  begin
    if n=0 then
      begin
        if ye[4]-p2old=0 then sigma:=0 else sigma:=(u-uold)/(ye[4]-p2old);
        n:=n+1; go to B
      end
      else
      begin
        n:=0; r:=0; h:=hin; go to C
      end
  end;
C:    if p=1 then
 begin
print(2,4,te); punch(8); print(2,2,u); punch(8);
for j:=1 step 1 until ntot do
  begin print(2,2,ye[j]); punch(0) end j;
punch(1);
end p
end RK1ST;
;
```

```
comment nu börjar programmet;
A1:
comment inläsning;
ntot:=read;
np:=read;
tin:=read;
hin:=read;
eps:=read;
delta:=read;
K:=read;
m:=read;
m1:=read;
b:=read;
for i:=1 step 1 until ntot do yin[i]:=read;
for i:=1 step 1 until np do slut[i]:=read;
A2:
a[1]:=yin[3]; a[2]:=yin[4];
antit:=1; p:=0; q:=1;
INITIAL:
print(3,0,antit); punch(1);
print(1,5,hin); punch(8);
for i:=1 step 1 until np do begin print(2,5,a[i]); punch(0) end;
punch(1);
UTSKRIFT:
h:=hin; t:=tin; s:=1;
for i:=1 step 1 until ntot do y[i]:=yin[i];
y[3]:=a[1]; y[4]:=a[2];
if p=1 then begin RK1ST(t,y,0,te,ye,p) end;
A3:
comment integrering;
RK1ST(t,y,h,te,ye,p);
if antit=1 then
  begin
   if ye[b]≥slut[b] then go to A42 else
     begin
      t:=te; for i:=1 step 1 until ntot do y[i]:=ye[i];
      go to A3
     end
  end;
if (norm[1]×(ye[1]-slut[1])+norm[2]×(ye[2]-slut[2]))≥0^s=1 then begin hin:=hin/m1;
h:=hin; s:=s+1; go to A3 end;
if norm[1]×(ye[1]-slut[1])+norm[2]×(ye[2]-slut[2])≥0 then go to A41 else
  begin
   t:=te; for i:=1 step 1 until ntot do y[i]:=ye[i];
   go to A3
  end;
A41:
comment utskrift av tiden och tillståndsvariablerna 1-8;
hin:=hin×m1;
A42:
if p=1 then go to OPTIMAL;
print(2,5,te); punch(1);
for i:=1 step 1 until ntot do begin print(2,2,ye[i]); punch(0); end;
punch(1);
;
```

```
A5:
comment undersökning av förlustfunktionen V;
V:=(ye[1]-slut[1])↑2+(ye[2]-slut[2])↑2;
print(2,5,V); punch(1);
if (q=1)^(V<delta) then begin q:=q+1; hin:=hin/2 end;
if V<eps then begin p:=1; punch(1); punch(1); go to UTSKRIFT end;
A6:
comment beräkning av normerade normalen p(T);
c1:=sqrt(ye[3]↑2+ye[4]↑2);
norm[1]:=ye[3]/c1;
norm[2]:=ye[4]/c1;
print(2,5,norm[1]); punch(0); print(2,5,norm[2]); punch(1);
A7:
comment beräkning av ändringen i normalen dp;
dp[1]:=-K×(ye[1]-slut[1]);
dp[2]:=-K×(ye[2]-slut[2]);
print(2,5,dp[1]); punch(0); print(2,5,dp[2]); punch(1);
A8:
comment uppdatering av normalen;
norm[1]:=norm[1]+dp[1];
norm[2]:=norm[2]+dp[2];
print(2,5,norm[1]); punch(0); print(2,5,norm[2]); punch(1);
A9:
comment beräkning av vektorn da;
det:=ye[5]×ye[8]-ye[6]×ye[7];
print(2,5,det); punch(1);
da[1]:=(ye[8]×dp[1]-ye[6]×dp[2])/det;
da[2]:=(ye[5]×dp[2]-ye[7]×dp[1])/det;
print(2,5,da[1]); punch(0); print(2,5,da[2]); punch(1);
A10:
comment beräkning av normerat nytt a;
a[1]:=a[1]+da[1];
a[2]:=a[2]+da[2];
c2:=sqrt(a[1]↑2+a[2]↑2);
a[1]:=a[1]/c2; a[2]:=a[2]/c2;
print(2,5,a[1]); punch(0); print(2,5,a[2]); punch(1);
A11:
antit:=antit+1; go to INITIAL;
OPTIMAL:
end
```

Användning av Pontryagins Maximumprincip på målsökningsproblem.

Examensarbete i regleringsteknik av Krister Mårtensson.

Målsökare har traditionellt dimensionerats med utgångspunkt från vissa s.k. styrprinciper t.ex. styrning utan felpekning, styrning med konstant framförhållning, styrning med syftbäringsprincipen. I detta arbete undersökes möjligheterna att i ett enkelt fall formulera målsökningsproblem som extremalproblem. I första etappen undersökes det fall då roboten på kortast möjliga tid skall träffa målet under förutsättningen att robotens tväracceleration är begränsad. I första hand studeras endast det plana fallet med förenklad robotdynamik. Ett nödvändigt villkor för den optimala styrningen erhålles i detta fall ur Pontryagins Maximumprincip. Villkoret är i form av ett randvärdesproblem för en ordinär differentialekvation. För att lösa detta problem skall en ALGOL algoritm utarbetas. Denna algoritm bör vara så flexibel att andra problem kan behandlas utan omfattande omprogrammering. Det ingår även i arbetet att orientera sig om optimeringsmetoder och robotstyrning.

1. Orientering om optimeringsmetoder.

Läs grunderna i

(1) Pontryagin, L.S. et al "The Mathematical Theory of Optimal Processes"

(2) Noton, A.R. "Introduction to Variational Methods in Control Engineering"

2. Numerisk lösning av optimeringsproblem.

Gör en kortfattad översikt av de metoder som finnes. Begränsa sedan arbetet till att iterera på initialvillkor.

(1) Bryson, A.E. "Optimal Programming and Control" IBM Scientific Computing Symposium, New York 1964

(2) Levine, M.D. Två bifogade artiklar

(3) Balakrishnan, A.V., Neustadt, L.W. "Computing Methods in Optimization Problems"

Utarbeta själv ett ALGOL program för lösning av Eulerekvationerna med hjälp av iteration på begynnelsevillkoren. Programmet bör vara flexibelt och uppbyggt av procedurer. Standardprocedurer för lösning av ekvationssystem och integration av ordinära differential-ekvationer tages ur litteraturen.

Skiss av programmet:

1. Gissa initialvillkor z(0)

2. Integrera Eulerekvationerna

$$\frac{dz}{dt} = f(z)$$

och störningsekvationerna

$$\frac{d}{dt}(\delta z) = f_z \, \delta z$$

3. Avbryt integrationen då slutvillkoret är uppfyllt

4. Undersök randvillkoren

5. Modifiera initialvillkoren så att felet i randvillkoren minskar.

6. Insätt de nya initialvillkoren och upprepa från 1.

Förslag till testexempel

Exempel: Betrakta systemet

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Bestäm en styrning sådan att systemet på kortast möjliga tid förflyttar sig från (0,0) till $(x^{\circ},0)$.

Hamiltonfunktionen lyder

$$\mathcal{H}(x,p,u) = p^T f(x,u) + 1 = p_1 x_2 + p_2 u + 1$$

Mina med avseende på u

$$u = -\operatorname{sgn} p_2$$

$$\frac{dp_1}{dt} = 0$$

$$\frac{dp_2}{dt} = p_1$$

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = u$$

Randvillkor:

$$x_1(0) = 0$$

$$x_2(0) = 0$$

$$x_1(T) = x^\circ$$

$$x_2(T) = 0$$

3. Orientering av robotstyrning.

Gör en litteraturöversikt. Räkna igenom följande fall:

Styrning av felpekning.

Antag att roboten styrs så att $\phi = \theta$ alltid. Integrera banekvationerna för detta fall och beräkna tiden och undersök den maximala accelerationen.

Styrning med konstant felpekning

Antag att roboten styrs så att

$$\phi - \theta = \phi(0) - \theta(0) = \text{konstant}$$

Integrera banekvationerna för detta fall, bestäm tiden till träff och undersök den maximala tväraccelerationen.

Syftbäringsstyrning (Syftbäringskonstant = 2)

Antag att roboten styrs så att följande samband gäller

$$\frac{d\phi}{dt} = C \cdot \frac{d\theta}{dt}$$

Bestäm banorna för C = 2. Undersök tiden till träff och den maximala tväraccelerationen.

4. Optimal robotstyrning

Välj parametrar för typfall med ledning av de data som erhölls under 3. Bestäm de optimala banorna och den optimala styrsignal. Beräkna syftbäringskonstant m.m.

Automatica, Vol. 3, pp. 203-217. Pergamon Press, 1966. Printed in Great Britain.

PAPER VI

TRAJECTORY OPTIMIZATION USING THE NEWTON-RAPHSON METHOD

M. D. LEVINE

1. INTRODUCTION

CONSIDERABLE work has been directed towards solving the two point boundary value problem which arises in the determination of optimal control functions. However, a great deal of research on this subject still remains to be done since sufficient experience with this type of problem has not been accumulated and many difficulties have yet to be sorted out. In particular, there is great hesitance about using the so-called boundary value iteration techniques although in practice these require the least computer time and storage facilities when compared with other more sophisticated methods. Also, for most problems these techniques are simpler to handle as well as to program.

In a previous paper [1], the author described such a method based on the concept of manipulating optimal trajectories by hillclimbing on the initial conditions of the adjoint equation. Although in many cases this procedure was considered to be adequate, the method described here provides much faster convergence. The latter uses a classical Newton–Raphson Method to perform the up-dating of the initial conditions for the successive iterations. Thus it does not require the programmer to guess at any constants save the initial values of the boundary conditions.

Two non-linear examples are presented and many of the difficulties which arise in doing this type of computation are discussed in detail.

2. FORMULATION OF THE PROBLEM

Given a dynamic system described by the vector differential equation

$$\dot{x}=f(x, u, t),\ x(t_0)=x_0 \tag{1}$$

where x is the n-dimensional state vector and u is the r-dimensional control vector. The problem is to choose the control function $u(t)$ over the time interval $t_0 \leq t \leq t_f$ to minimize the cost functional

$$P(u)=\int_{t_0}^{t_f} f_0(x, u, t)\ \mathrm{d}t \tag{2}$$

and to satisfy the terminal conditions

$$S_j[x(t_f), t_f]=0, j=1, \ldots q<n+1. \tag{3}$$

[1] M. D. LEVINE: *A Steepest Descent Method for Synthesizing Optimal Control Programmes*, presented to the Inst. of Mech. Engineers' Convention on 'Advances in Automatic Control', University of Nottingham, April (1965).

If a Hamiltonian function H is defined as

$$H=-f_0+pf \tag{4}$$

where p is the n-dimensional adjoint vector, then PONTRYAGIN'S Maximum Principle [2] requires that for optimality:

$$\dot{x}=\frac{\partial H}{\partial p}=f \tag{5}$$

$$\dot{p}=\frac{-\partial H}{\partial x}=g \tag{6}$$

$$\frac{\partial H}{\partial u}=0 \tag{7}$$

$$[H\,\mathrm{d}t-p\mathrm{d}x]_{t=t_f}=0 \tag{8}$$

The control u may be determined as a function of x, p, and t by invoking equation (7): substituting into equations (5) and (6), the following set of $2n$ differential equations may be derived:

$$\dot{x}=f(x, p, t) \tag{9}$$

$$\dot{p}=g(x, p, t). \tag{10}$$

The transversality condition of equation (8) applied to equation (3) yields $n+1$ relationships which must be satisfied at $t=t_f$. Hence

$$\phi_j(x, p, t)=0, j=1, \ldots, n+1. \tag{11}$$

The vectors x and p (referred to as the canonical variables) plus the independent variable t constitute $2n+1$ variables. The $2n+1$ boundary values are specified by the n initial conditions of the state equations, x_0, and the $n+1$ equations (11). Since half these conditions are given at $t=t_0$ and the other half at $t=t_f$, this is a two point boundary value problem.

To perform the integration of equations (9) and (10), it is necessary to have an integration stopping function. It is convenient to choose a function W such that

$$W(x, p, t)-\phi_m(x, p, t)=0,\ 1\leq m\leq n+1. \tag{12}$$

Since the above equation will always be satisfied at $t=t_f$, only n terminal conditions of the type described by equation (11) remain to be satisfied.

The problem may now be stated as follows: Find the value of the n-dimensional vector

$$p(t_0)=a \tag{13}$$

such that when equations (9) and (10) are integrated until $t=t_f$ defined by equation (12), the n relations of equation (6) will also be satisfied. The solution of equation (9) is then the required optimal trajectory in phase space.

[2] L. S. PONTRYAGIN, V. G. BOLTYANSKII, R. V. GAMKRELIDZE and E. F. MISHCHENKO: *The Mathematical Theory of Optimal Processes.* Interscience, New York (1962).

3. THE COMPUTATIONAL ALGORITHM

In order to derive a digital computation scheme, define a vector Ψ such that

$$\Psi_i=\phi_i^2, \qquad i=1, \ldots, n \tag{14}$$

If the optimum value of the initial condition of the adjoint equation is referred to as a^*, it is obvious that

$$\Psi_i>0 \text{ for } a\neq a^*$$
$$\Psi_i=0 \text{ for } a=a^*, \; i=1, \ldots, n. \tag{15}$$

Consequently, if it is assumed that

$$\Psi_1(a_1, a_2, \ldots, a_n)=0$$
$$\Psi_2(a_1, a_2, \ldots, a_n)=0$$
$$\vdots$$
$$\Psi_n(a_1, a_2, \ldots, a_n)=0 \tag{16}$$

then it follows that the value of a which satisfies these equations is $a=a^*$.

Using a Taylor Series expansion for the above n equations and discarding all terms of order greater than one, it can be shown that

$$a_{new}=a_{old}-\left[\frac{\partial\Psi}{\partial a}\right]^{-1}\Psi. \tag{17}$$

This linear approximation is the Newton–Raphson Method in n dimensions. It is interesting to note that while here the correction is inversely proportional to the slope, in general, hillclimbing schemes up-date by correcting by an amount directly proportional to the slope.

Differentiating equations (14) with respect to a yields

$$\frac{\partial\Psi}{\partial a}=2\phi\left[\frac{\partial\phi}{\partial x}\frac{\partial x}{\partial a}+\frac{\partial\phi}{\partial p}\frac{\partial p}{\partial a}+\frac{\partial\phi}{\partial t_f}\frac{\partial t_f}{\partial a}\right] \tag{18}$$

Define sensitivity matrices $Y=\partial x/\partial a$ and $Z=\partial p/\partial a$. It can be shown that [1] these functions may be determined along any trajectory by the integration of the following accessory equations:

$$\frac{dY}{dt}=\frac{\partial f}{\partial x}Y+\frac{\partial f}{\partial p}Z, \; Y(t_0)=0 \tag{19}$$

$$\frac{dZ}{dt}=\frac{\partial g}{\partial p}Z+\frac{\partial f}{\partial x}Y, \; Z(t_0)=I \tag{20}$$

Also, with the aid of equation (7)

$$\frac{\partial t_f}{\partial a}=\frac{\left[\frac{\partial W}{\partial x}Y+\frac{\partial W}{\partial p}Z\right]}{\frac{\partial W}{\partial x}f+\frac{\partial W}{\partial p}g+\frac{\partial W}{\partial t}}. \tag{21}$$

Hence $\partial\Psi/\partial a$ in equation (18) can be calculated for $W=0$ with a knowledge of x, p, t_f, Y and Z at $t=t_f$.

The computational algorithm may be outlined as follows:

1. Guess a value of a.
2. Integrate equations (9), (10), (19) and (20) until $W(t_f)=0$.
3. Evaluate $V=\sum_{i=1}^{m}\Psi_i$. If $V\leq\varepsilon$, where ε is some predetermined small positive number, the optimal $a=a^*$ has been found.
4. If $V>\varepsilon$, up-date the value of a according to equation (18).
5. Return to step 2.

Examples demonstrating the use of this technique are presented in the next two sections.

4. THE TUBULAR REACTOR DESIGN PROBLEM

For this problem, it is required to determine the optimum temperature gradient in a chemical reactor [3]. The order of the reactor is $A\rightarrow B\rightarrow C$ where the product of interest is B. Both reactions are assumed to be of the first order.

Let the concentrations of A and B be x_1 and x_2 respectively. The nonlinear dynamics of the reactions may be described by the following equations:

$$\dot{x}_1=-k_1x_1,\ x_1(0)=x_{10} \tag{22}$$

$$\dot{x}_2=k_1x_1-k_2x_2,\ x_2(0)=x_{20} \tag{23}$$

where

$$k_1=G_1\exp\left[\frac{-E_1}{Ru}\right],\qquad k_2=G_2\exp\left[\frac{-E_2}{Ru}\right]$$

are the rate constants of the reactions. The independent variable is t which is the holding time of the reactor up to a given point. The temperature at that point is $u(t)$ which is treated as the control function in this problem. The constants in the above equations are chosen as:

$$G_1=0{\cdot}535\times10^{11}\ \text{min}^{-1}$$

$$G_2=0{\cdot}461\times10^{18}\ \text{min}^{-1}$$

$$E_1=18000\ \text{cal./mole}$$

$$E_2=30000\ \text{cal./mole}$$

[3] E. S. LEE: *Optimization by Pontryagin's Maximum Principle on the Analog Computer*. 1963 Joint Automatic Control Conference.

$R=2$ cal./mole$-$°K

$x_{10}=0{\cdot}53$ mole/l.

$x_{20}=0{\cdot}43$ mole/l.

The problem is to maximize the yield of B defined as x_2, over the total holding time $t=t_f$ where $t_f=8$ min.

By applying equations (5), (6) and (7), the control is found to be

$$u=\left[\frac{E_2-E_1}{R}\right]\ln\alpha \tag{24}$$

where

$$\alpha=\left[\frac{E_2G_2}{E_1G_1}\right]\left[\frac{p_2x_2}{(p_2-p_1)x_1}\right];$$

the adjoint equations become

$$\dot{p}_1=k_1(p_1-p_2),\ p_1(0)=a_1 \tag{25}$$

$$\dot{p}_2=k_2p_2,\ p_2(0)=a_2. \tag{26}$$

The transversality condition of equation (8) yields the requirement that

$$\phi_1=p_1(t_f) \tag{27}$$

$$\phi_2=p_2(t_f)-1. \tag{28}$$

Thus it is necessary to define

$$\Psi_1(a_1, a_2)=p_1^2 \tag{29}$$

$$\Psi_2(a_1, a_2)=(p_2-1)^2. \tag{30}$$

Since a Newton–Raphson Method was used to solve this problem, a_1 and a_2 were up-dated by invoking equation (17) for the two dimensional case:

$$a_{1\text{new}}=a_{1\text{old}}-\frac{1}{\Delta}\left[\frac{\partial\Psi_2}{\partial a_2}\Psi_1-\frac{\partial\Psi_1}{\partial a_2}\Psi_2\right] \tag{31}$$

$$a_{2\text{new}}=a_{2\text{old}}-\frac{1}{\Delta}\left[-\frac{\partial\Psi_2}{\partial a_1}\Psi_1+\frac{\partial\Psi_1}{\partial a_1}\Psi_2\right] \tag{32}$$

where

$$\Delta=\frac{\partial\Psi_1}{\partial a_1}\frac{\partial\Psi_2}{\partial a_2}-\frac{\partial\Psi_2}{\partial a_1}\frac{\partial\Psi_1}{\partial a_2}$$

$$\frac{\partial\Psi_1}{\partial a_i}=2p_1Z_{1i},\qquad i=1, 2$$

$$\frac{\partial\Psi_2}{\partial a_i}=2(p_2-1)Z_{2i},\qquad i=1, 2.$$

The variables Z_{1i} and Z_{2i} are calculated by integrating the accessory equations. These are given in Appendix A.

An integration step length of $t=0{\cdot}08$ was chosen and the integration stopping condition was obviously

$$W=t-8. \tag{33}$$

A fourth order Runge-Kutta integration method was used. The iterations were begun with initial guesses of $a_1=2$ and $a_2=3$. It took 14 iterations, 4 sec on the London Atlas Computer, to reduce the magnitude of $V=\Psi_1+\Psi_2$ below $\varepsilon=10^{-6}$. The optimal values of the initial conditions of the adjoint equation were found to be $a_1=0{\cdot}610126$ and $a_2=0{\cdot}828495$. The optimum temperature profile is shown in Fig. 1; the concentrations are plotted as a function of t in Fig. 2; the solution of the accessory equations along the optimal trajectory

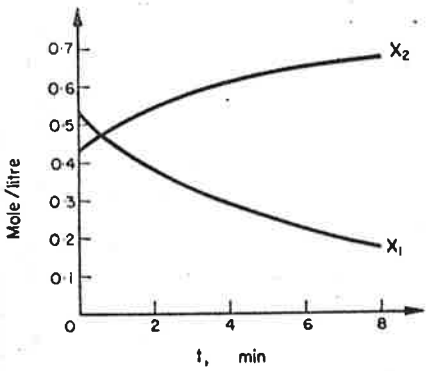


FIG. 1. Reactor concentrations.

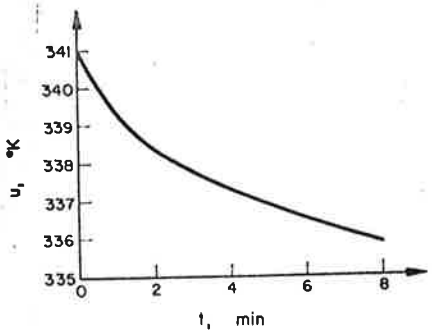


FIG. 2. Temperative profile.

has been plotted in Figs. 3a, b. For this problem it was found that the Method of Steepest Descent as described in Reference [1] converged considerably slower than the method described here.

One of the arguments often put forth against an initial condition variation technique is that a good estimate of a is required to achieve reasonable convergence. This does not seem to be a valid point for the general case. In fact, depending on the problem, this may or may not hold true. It does not seem possible to predict this before the actual computations are performed.

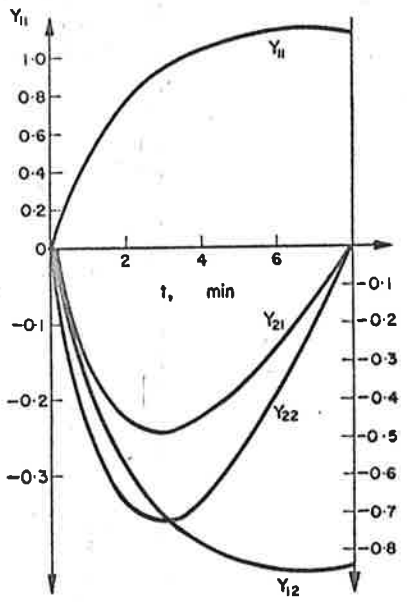


FIG. 3a. The accessory variable Y.

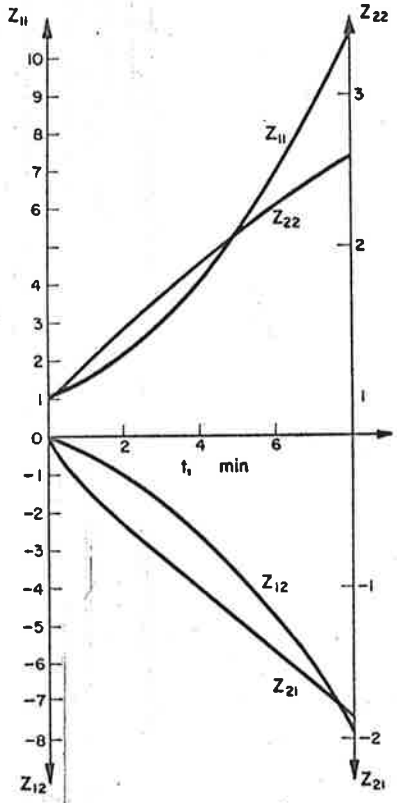


FIG. 3b. The accessory variable Z.

To investigate the performance of this method, the Tubular Reactor problem was solved for varying starting values of a. The results are given in Table 1. Each iteration took about $\frac{1}{4}$ sec computing time on the Atlas. It is interesting to note that although the value of a_1 and a_2 varied slightly from test to test, the values of $x_1(8)$ and $x_2(8)$ remained unaltered. In general, the farther from the optimal solution the problem was commenced, the more iterations were required. Thus for this nonlinear problem the method used converged without difficulty to the solution with the initial guesses well off the correct value.

TABLE 1. EXPERIMENTAL CALCULATIONS FOR THE TUBULAR REACTOR PROBLEM.

initial a_1	initial a_2	a_1^*	a_2^*	no. of iterations	$x_1(8)$	$x_2(8)$
2	3	0·610126	0·828495	14	0·1704	0·6794
2×10^1	3×10^1	0·610248	0·828673	17	0·1704	0·6794
2×10^2	3×10^2	0·610155	0·828519	21	0·1704	0·6794
2×10^3	3×10^3	0·610198	0·828589	24	0·1704	0·6794
2×10^4	3×10^4	0·610251	0·828675	27	0·1704	0·6794
2×10^5	3×10^5	0·610151	0·828514	31	0·1704	0·6794
2×10^6	3×10^6	0·610193	0·828581	34	0·1704	0·6794
-2	-3	0·609658	0·827909	15	0·1703	0·6794
-2×10^1	-3×10^1	0·609645	0·827898	18	0·1703	0·6794
-2×10^2	-3×10^2	0·609530	0·827625	20	0·1704	0·6794

5. THE INTERCEPTION PROBLEM

This problem is described in Reference [4]. It is required to choose the thrust program $u(t)$ for a rocket which is to intercept and match the speed of an orbiting vehicle travelling on a known path. In addition, it is desirable that the rocket should achieve contact with the burning of a minimum amount of fuel.

The state equations for the motion in a single dimension are

$$\dot{x}_1=\frac{-9\cdot8}{(1+x_2)^2}+(1+e^{-10t})^{-1}\left[10x_1e^{-10t}-\frac{(2x_1)^7}{(1+10x_2)^8}+u\right],x_1(0)=0 \tag{34}$$

$$\dot{x}_2=x_1,\ x_2(0)=0 \tag{35}$$

where x_1 and x_2 are the rocket velocity and position, respectively. The orbiting vehicle follows the trajectory

$$x_1(t)=2t-1 \tag{36}$$

$$x_2(t)=t^2-t+0\cdot35 \tag{37}$$

in phase space. The control $u(t)$ must be chosen to minimize

$$P(u)=\int_{t_0}^{t_f}u^2\,dt. \tag{38}$$

[4] W. KIPINIAK: *Dynamic Optimization and Control.* MIT Press and Wiley, New York (1961).

Invoking equations (5), (6) and (7), the control for the optimal trajectory is found to be

$$u(t)=0{\cdot}5p_1 \tag{39}$$

where the adjoint equations are:

$$\dot{p}_1=(1+e^{+10t})^{-1}\left[\frac{14p_1(2x_1)^6}{(1+10x_2)^8}-p_2\right], \qquad p_1(0)=a_1 \tag{40}$$

$$\dot{p}_2=-p_1\left[\frac{19{\cdot}6(1+e^{-10t})}{(1+x_2)^3}+\frac{80(2x_1)^7}{(1+10x_2)^9}\right], \qquad p_2(0)=a_2. \tag{41}$$

Using the transversality condition of equation (8), the terminal conditions become

$$\phi_1=x_2-t^2+t-0{\cdot}35 \tag{42}$$

$$\phi_2=p_1(x_1-2)+p_2(x_2-2t+1)-0{\cdot}25{p_1}^2 \tag{43}$$

$$\phi_3=x_1-2t+1. \tag{44}$$

The stopping condition is chosen as

$$W=\phi_3 \tag{45}$$

so that the following implicit equations remain:

$$\Psi_1(a_1, a_2)={\phi_1}^2=0 \tag{46}$$

$$\Psi_2(a_1, a_2)={\phi_2}^2=0. \tag{47}$$

As was done in section 4, the Newton–Raphson Method was used to up-date a_1 and a_2. Equation (17) requires $\partial\Psi/\partial a$: referring to equations (42), (43), (46) and (47) yields:

$$\frac{\partial\Psi_1}{\partial a_i}=2\phi_1[Y_{2i}+(0{\cdot}5-t)Y_{1i}], \qquad i=1, 2 \tag{48}$$

$$\frac{\partial\Psi_2}{\partial a_i}=2\phi_2[Z_{1i}(\dot{x}_1-2)+p_1G_i+Z_{2i}(\dot{x}_2-2t+1)-0{\cdot}5p_1Z_{1i}], \qquad i=1, 2 \tag{49}$$

where

$$G_i=(A_{11}+0{\cdot}5G_3)Y_{1i}+A_{12}Y_{2i}+0{\cdot}5(1+e^{-10t})^{-1}Z_{1i}, \; i=1, 2$$

$$G_3=\left[\frac{10e^{-10t}}{(1+e^{-10t})^2}\right]\left[10x_1e^{-10t}\frac{(2x_1)^7}{(1+10x_2)^8}+u\right]-\left[\frac{100x_1e^{-10t}}{1+e^{-10t}}\right]. \tag{50}$$

Again to evaluate Y and Z, it is necessary to integrate the accessory equations. These are given in Appendix B for the canonical equations (34), (35), (40) and (41).

The computations were begun with starting values $a_1=40$ and $a_2=250$. The integration step length was chosen as $t=0{\cdot}001$. To reduce $V=\Psi_1+\Psi_2$ to $\varepsilon=10^{-6}$ required 45 iterations —this took 28 sec on the Atlas Computer. The results were:

$$a_1^* = 47{\cdot}2609$$
$$a_2^* = 320{\cdot}242$$
$$t_f = 0{\cdot}399118 \text{ sec}$$
$$x_1(t_f) = 0{\cdot}202029$$
$$x_2(t_f) = 0{\cdot}110113.$$

The trajectory in phase space, the control function, and the accessory variables along the optimal trajectory are plotted in Figs. 4, 5, 6a and 6b respectively. Since the above accuracy is not usually required, the program could have been terminated when $V \leq \varepsilon$, $\varepsilon = 10^{-2}$ which would have given after 26 iterations

$$a_1^* = 47{\cdot}2647$$
$$a_2^* = 320{\cdot}312$$
$$t_f = 0{\cdot}398985 \text{ sec}$$
$$x_1(t_f) = 0{\cdot}201762$$
$$x_2(t_f) = 0{\cdot}110176.$$

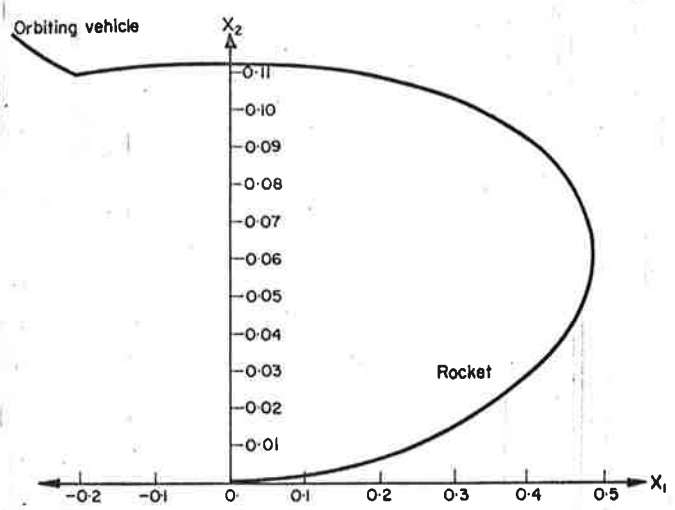


FIG. 4. Rocket trajectory in phase space.

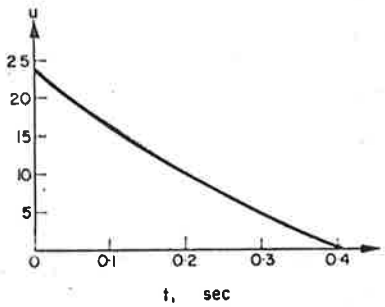


FIG. 5. Thrust programme.

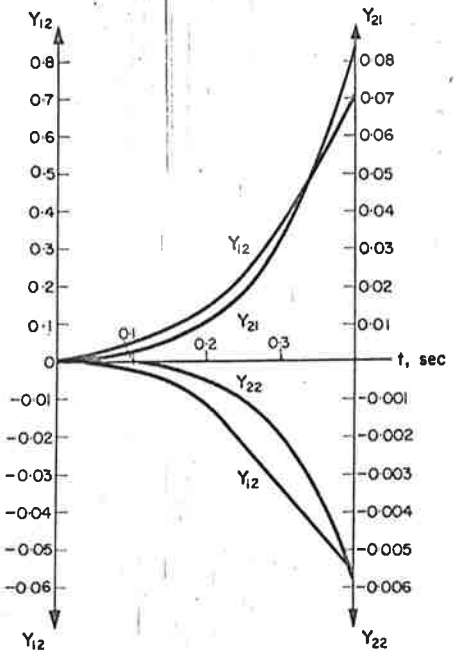


FIG. 6a. The accessory variable Y.

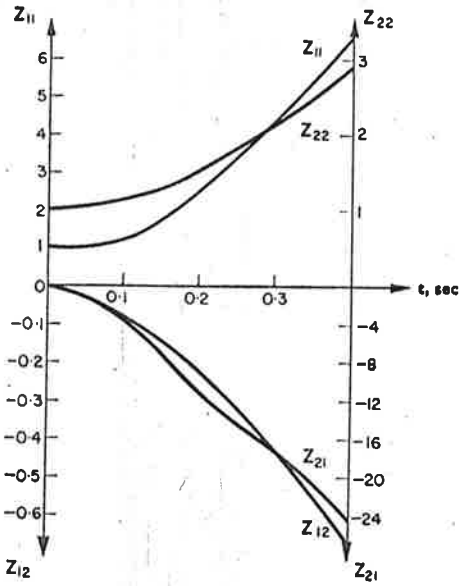


FIG. 6b. The accessory variable Z.

The final time in this case differs only in the fourth decimal place. This brings up an important practical point: it is difficult to decide *a priori* on a magnitude of ε. In the above problem very little is gained by running the program for the extra 12 sec.

As opposed to the Tubular Reactor Problem, in this case the adjoint equations (40) and (41) became unstable for values of a different from a^* by more than about 25%. Consequently, without any approximate knowledge of a^*, a preliminary computer search would be necessary to determine a suitable neighbourhood of the optimal trajectory.

6. CONCLUSIONS

A method for solving the two point boundary value problems using the Newton–Raphson iteration technique has been described. One of the main advantages of employing this method for determining the optimal trajectory is that it usually requires less computer storage and time than other techniques such as those based on the successive approximation of the control function [5].

It has been shown that it is difficult to discern at the outset how stable the adjoint equations will be for a given guess of the initial conditions. This may or may not be of consequence, depending on the problem. Probably the most fruitful procedure for the general case would be one which used a crude successive approximation technique to find a suitable neighbourhood of the optimum trajectory and then switched over a method of the type described here.

APPENDIX A

Accessory Equations for the Tubular Reactor Problem.

In order to utilize the technique described, it is necessary to determine the sensitivity matrices.

$$Y=\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \quad \text{and} \quad Z=\begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

These may be evaluated for the Tubular Reactor Problem by integrating the following accessory equations:

$$\dot{Y}=AY+BZ,\ Y(0)=0$$

$$\dot{Z}=CZ+DY,\ Z(0)=\mathrm{I}$$

where the matrices A, B, C, D are defined as

[5] H. J. KELLEY: Method of Gradients, in: *Optimisation Techniques*, Chap. 6, ed. by G. LEITMANN. Academic Press, New York (1962).

$$A=\begin{bmatrix} -k_1-x_1\dfrac{\partial k_1}{\partial x_1} & -x_1\dfrac{\partial k_1}{\partial x_2} \\ k_1+x_1\dfrac{\partial k_1}{\partial x_1}-x_2\dfrac{\partial k_2}{\partial x_1} & x_1\dfrac{\partial k_1}{\partial x_2}-k_2-x_2\dfrac{\partial k_2}{\partial x_2} \end{bmatrix}$$

$$B=\begin{bmatrix} -x_1\dfrac{\partial k_1}{\partial p_1} & -x_2\dfrac{\partial k_1}{\partial p_2} \\ x_1\dfrac{\partial k_1}{\partial p_1}-x_2\dfrac{\partial k_2}{\partial p_1} & x_1\dfrac{\partial k_1}{\partial p_2}-x_2\dfrac{\partial k_2}{\partial p_2} \end{bmatrix}$$

$$C=\begin{bmatrix} k_1+(p_1-p_2)\dfrac{\partial k_1}{\partial p_1} & -k_1+(p_1-p_2)\dfrac{\partial k_1}{\partial p_2} \\ p_2\dfrac{\partial k_2}{\partial p_2} & \dfrac{\partial k_2}{p_2\partial p_2}+k_2 \end{bmatrix}$$

$$D=\begin{bmatrix} (p_1-p_2)\dfrac{\partial k_1}{\partial x_1} & (p_1-p_2)\dfrac{\partial k_1}{\partial x_2} \\ p_2\dfrac{\partial k_2}{\partial x_1} & p_2\dfrac{\partial k_2}{\partial x_2} \end{bmatrix}$$

Also

$$\frac{\partial k_i}{\partial p_j}=\frac{\partial k_i}{\partial \alpha}\frac{\partial \alpha}{p_j}, \qquad i=1, 2; j=1, 2$$

$$\frac{\partial k_i}{\partial x_j}=\frac{\partial k_i}{\partial \alpha}\frac{\partial \alpha}{x_j}, \qquad i=1, 2; j=1, 2$$

where

$$\frac{\partial k_1}{\partial \alpha}=\frac{1{\cdot}5k_1}{\alpha}$$

$$\frac{\partial k_2}{\partial \alpha}=\frac{-2{\cdot}5k_2}{\alpha}$$

$$\frac{\partial \alpha}{\partial p_1}=\left(\frac{E_2G_2}{E_1G_1}\right)\left[\frac{p_1x_2}{(p_2-p_1)^2x_1}\right]$$

$$\frac{\partial\alpha}{\partial p_2}=\left(\frac{E_2G_2}{E_1G_1}\right)\left[\frac{x_2}{(p_2-p_1)x_1}-\frac{p_2x_2}{(p_2-p_1)^2x_1}\right]$$

$$\frac{\partial\alpha}{\partial x_1}=\left(\frac{E_2G_2}{E_1G_1}\right)\left[\frac{p_2x_2}{(p_2-p_1)^3x_1^2}\right]$$

$$\frac{\partial\alpha}{\partial x_2}=\left(\frac{E_2G_2}{E_1G_1}\right)\left[\frac{p_2}{(p_2-p_1)x_1}\right]$$

APPENDIX B

Accessory Equations for the Interception Problem

The accessory equations for the canonical equations (34), (35), (40) and (41) of the Interception Problem are

$$\dot{Y}_{11}=A_{11}Y_{11}+A_{12}Y_{21}+0{\cdot}5B_{11}Z_{11},\ Y_{11}(0)=0$$

$$\dot{Y}_{12}=A_{11}Y_{12}+A_{12}Y_{22}+0{\cdot}5B_{11}Z_{12},\ Y_{12}(0)=0$$

$$\dot{Y}_{21}=Y_{11},\ Y_{21}(0)=0$$

$$\dot{Y}_{22}=Y_{12},\ Y_{22}(0)=0$$

$$\dot{Z}_{11}=C_{11}Z_{11}-B_{11}Z_{21}+D_{11}Y_{11}+D_{12}Y_{21},\ Z_{11}(0)=1$$

$$\dot{Z}_{12}=C_{11}Z_{12}-B_{11}Z_{22}+D_{11}Y_{12}+D_{12}Y_{22},\ Z_{12}(0)=0$$

$$\dot{Z}_{21}=C_{21}Z_{11}+D_{21}Y_{11}+D_{22}Y_{21},\ Z_{21}(0)=0$$

$$\dot{Z}_{22}=C_{21}Z_{12}+D_{21}Y_{12}+D_{22}Y_{22},\ Z_{22}(0)=1$$

where

$$A_{11}=\left[10e^{-10t}-\frac{14(2x_1)^6}{(1+10x_2)^8}\right]B_{11}$$

$$A_{12}=\frac{19{\cdot}6}{(1+x_2)^3}+\left[\frac{80(2x_1)^7}{(1+10x_2)^9}\right]B_{11}$$

$$B_{11}=(1+e^{-10t})^{-1}$$

$$C_{11}=\frac{14B_{11}(2x_1)^6}{(1+10x_2)^8}$$

$$C_{21}=\frac{-19.6}{B_{11}(1+x_2)^3}-\frac{80(2x_1)^7}{(1+10x_2)^9}$$

$$D_{11}=\left[\frac{168(2x_1)^5}{(1+10x_2)^8}\right]p_1B_{11}$$

$$D_{12}=-\left[\frac{112(2x_1)^6}{(1+10x_2)^9}\right]p_1B_{11}$$

$$D_{21}=\frac{10D_{12}}{B_{11}}$$

$$D_{22}=\left[\frac{58{\cdot}8}{B_{11}(1+x_2)^4}+\frac{7200(2x_1)^7}{(1+10x_2)^{10}}\right]p_1$$

Acknowledgements—The author would like to thank the Ministry of Youth, Province of Quebec and the National Research Council, Ottawa for the financial support which made this research possible.

K J Åström

Paper 4

A STEEPEST DESCENT METHOD FOR SYNTHESIZING OPTIMAL CONTROL PROGRAMMES

By M. D. Levine*

INTRODUCTION

THE BASIC CONTROL ENGINEERING PROBLEM involves the operation of a given system in some optimal or best fashion. Since most systems are subject to both internal and external random disturbances it is necessary to make a proper identification of the system and subsequently take these disturbances into account when devising the controller. Thus it is possible to consider two separate problems:

(i) the determination of the optimal control, and
(ii) the identification of the system parameters.

Very generally, we may depict this type of control system in a block diagram as shown in Fig. 4.1.

The second problem will not be considered here: an interesting survey may be found in (1)†. It will be assumed throughout that all variables and parameters are deterministic. This in effect assumes the identification problem has been solved. Thus, at any given time during the operation of the system, it is required to synthesize a control programme for a given time interval in the future. This is accomplished by finding the solution to a two-point boundary value problem, using a method based on the manipulation of the initial conditions of adjoint equations.

Note that in all probability the controller in Fig. 4.1 would have to be a special purpose digital computer which would sample the system and perform the computations involved in the calculation of the control function. The identification of the system parameters could also be done by the same machine. Thus, in the figure, the blocks enclosed by the broken line could be replaced by a single block representing a digital computer.

The MS. of this paper was first received at the Institution on 18th June 1964 and in its revised form, as accepted by the Council for publication, on 10th November 1964.
* *Electrical Engineering Department, Imperial College, London.*
† *References are given in Appendix 4.IV.*

Notation

A vector matrix notation is used throughout. No attempt has been made to differentiate between row and column vectors as the usage is obvious from the context. Partial derivatives are shown as subscripts: for example, f_x.

a	Vector initial condition of the adjoint equation.
$C(x, u, t)$	Vector control constraint.
c	Vector 'slack' variable.
E^*	Optimal trajectory.
e	Lagrange multiplier.
F	Lagrangian.
f	Vector function specifying the derivative of the state variable.
g	Vector function specifying the derivative of the adjoint variable.
H	Hamiltonian.
h	Integration step length.
k	Hill climbing constant.
$\bar{P}$	Set of all possible values of the adjoint variable.
$P(u)$	Performance criterion.
p	Adjoint vector.
S	Vector function prescribing the terminal conditions of the trajectory in phase space.
T	Set of all time instants.
t	Time.
U	Set of all possible controls.
u	Control vector.
V	Terminal computation error criterion.
W	Scalar function determining the stopping condition for the integration.
X	Set of all possible states.
x	State vector.
Y	Matrix accessory variable, x_a.
Z	Matrix accessory variable, p_a.
γ_j	Time at which a control discontinuity occurs.
ϕ	Vector function specifying the terminal boundary conditions of the canonical equations.

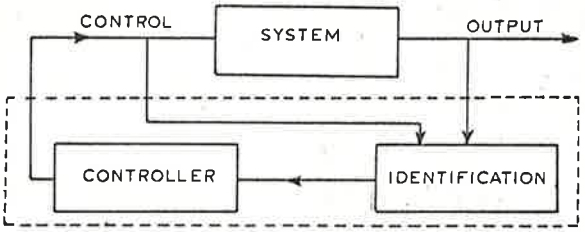


Fig. 4.1. Block diagram of a control system

THE CONTROL PROBLEM

The dynamic system in Fig. 4.1 may be described by the following differential equation:

$$\frac{dx}{dt} = f(x, u, t) \quad . \quad . \quad . \quad . \quad (1)$$

where x is an n-dimensional state vector, u is an r-dimensional control vector, and t is the independent variable time. Let f be of class C^1 in x and u, and of class D^0 in t. It is assumed that $x \in X$ and is continuous in $t \in T$, where X is the state space and t is the set of values of time at which the behaviour of the system is defined. Also $u \in U$, where U, the control space, is defined as the set of all possible values which $u(t)$ can assume at any time $t \in T$.

The set defining U may be described by

$$C(x, u, t) \geqslant 0 \quad . \quad . \quad . \quad . \quad (2)$$

where C is a k-dimensional vector. Equation (2) describes a state-dependent control constraint. In some cases C is a function of u and t only and so the set U is independent of the state x. Note that C must be such that if $k > r$, then at most only r components of C can vanish at any time t. To ensure that the constraints are compatible, it is necessary at each t for the matrix C_u taken over all $C = 0$ to have maximum rank. If $u \in U$ and is of class D^0 it shall be referred to as an admissible control.

The optimization of the system requires the specification of a performance index or cost function, $P(u)$, which must be extremized (maximized or minimized) by the proper choice of the control function $u(t)$. Thus define

$$P(u) = G_1[x(t_f), t_f] + \int_{t_0}^{t_f} G_2(x, u, t)\, dt \qquad (3)$$

where G_1 and G_2 are scalar functions. Assume that $P(u)$ is of class C^1 in x and u, and of class D^0 in t. Equation (3) is the so-called formulation of the Problem of Bolza (2). It will be found convenient to convert every Problem of Bolza into an equivalent Problem of Lagrange where the cost function is of the form

$$P(u) = \int_{t_0}^{t_f} f_0(x, u, t)\, dt \quad . \quad . \quad . \quad (4)$$

Equation (3) may be easily transformed into one of the form of equation (4) by defining an additional state equation x_{n+1} (3). Let

$$\frac{dx_{n+1}}{dt} = 0; \qquad x_{n+1}(t_f) = \frac{G_1[x(t_f), t_f]}{t_f - t_0} \quad . \quad (5)$$

then

$$f_0 = G_2(x, u, t) + x_{n+1}(t) \quad . \quad . \quad . \quad (6)$$

In general it will be assumed that the initial state $x(t_0)$ and the initial time t_0 are given. However, it will also be required that the trajectory of $x(t)$ in the phase space $X \times T$ satisfies the following terminal conditions:

$$S_j[x(t_f), t_f] = 0, \quad j = 1, \ldots, q < n+1 \qquad (7)$$

If t_f is known, then $q < n$. This is the problem with variable endpoints. The q equations in equation (7) define q hypersurfaces in the phase space. Consequently if the rank of the matrix $[S_t \;\vdots\; S_x]$ is q, then the intersection of these surfaces forms an $(n-q)$-dimensional terminal manifold, M_{t_f}. It is required that $[x(t_f), t_f] \in M_{t_f}$.

The basic control problem which will be treated is now stated: Find the control $u \in U$ which extremizes

$$P(u) = \int_{t_0}^{t_f} f^0(x, u, t)\, dt$$

for the system described by the dynamics $dx/dt = f(x, u, t)$, $x \in X$, given the initial condition in phase space, $[x(t_0), t_0]$, and the requirement that x must terminate on M_{t_f}.

In order to synthesize the optimal control programme $u(t)$, the Theory of the Calculus of Variations must be invoked. The necessary conditions for a *maximum* of $P(u)$ are well known (3) (4) (5) (6) (7); Hestenes (7) seems to have been the first person to solve this type of problem using the formulation of the Calculus of Variations. With this approach it is possible to outline the necessary conditions for an optimal control and subsequently derive the following equations (*see* Appendix 4.I):

$$\frac{dx}{dt} = f(x, p, t) \quad . \quad . \quad . \quad . \quad (8)$$

$$\frac{dp}{dt} = g(x, p, t) \quad . \quad . \quad . \quad . \quad (9)$$

The vector p is the so-called adjoint variable and is defined in Appendix 4.I.

There are $(2n+1)$ variables and so $(2n+1)$ boundary conditions are required. The n initial conditions for equation (8), $x(t_0)$, are presumed given, and equations (7) and (59) provide the $(n+1)$ additional conditions. Note that half the conditions are known at $t = t_0$ and the other half at $t = t_f$. The latter may be described by

$$\phi_j(x, p, t) = 0, \quad j = 1, \ldots, n+1 \quad . \quad (10)$$

The computational scheme to be described later requires a stopping condition to terminate the integration of equations (8) and (9); thus it is necessary to define a function W such that the integration is ended when

$$W(x, p, t) = 0 \quad . \quad . \quad . \quad . \quad (11)$$

It is convenient to choose one of the ϕ_j in equation (10) as the stopping condition. Hence,

$$W(x, p, t) = \phi_m(x, p, t) = 0 \quad . \quad . \quad (12)$$

if the mth terminal constraint is used. A practical choice for ϕ_m is one which explicitly involves the independent

THE COMPUTATIONAL TECHNIQUE

Using the terminal conditions of equation (13) it is possible to construct a scalar terminal error criterion, V. Define

$$V(t_f) = V[x(t_f, a), p(t_f, a), t_f(a)] = \sum_{i=1}^{n} [\phi_i(t_f)]^2 \quad (14)$$

Note that $V(t_f)$ may be an explicit function of x, p, t_f but is an implicit function of t_f and a. It is a positive semi-definite function with a minimum value of zero. For example, in a two-dimensional problem a_1 and a_2 are considered as the co-ordinates of a plane and $V(t_f)$ is plotted on the vertical axis. Then the criterion function will appear as a 'valley' in an $(n+1)$-dimensional hyperspace as shown in Fig. 4.2.

It is possible to assume a value for the vector a and then integrate equations (8) and (9) until $W = 0$. This defines a $t = t_f$ for which $V(t_f)$ may be determined from equation (14). This value may then be plotted as a point on the error hypersurface. Obviously, in order to solve the problem, it is necessary to determine the value of a which makes $V(t_f)$ vanish. This can be accomplished by choosing an initial point on the hypersurface and then descending the 'valley' in small steps by moving in the direction of steepest descent. Hence, the gradient of the hypersurface at any point on it is required; in other words, for a given a, a knowledge of V_a is desired. Choosing

$$\frac{da}{d\sigma} = -kV_a \quad . \quad . \quad . \quad . \quad (15)$$

where σ is a parameter and k is a constant, ensures continuous descent of the hill. Because of the discrete nature of the digital computation, equation (15) must be rewritten as

$$a_{\text{new}} = a_{\text{old}} - kV_a(t_f) \quad . \quad . \quad . \quad (16)$$

Now

$$V_a(t_f) = V_x(t_f)x_a(t_f) + V_p(t_f)p_a(t_f) + V_{t_f}(t_f)t_{f_a} \quad (17)$$

where the functions V_x, V_p, and V_{t_f} may be found directly by differentiation. It is still necessary to derive values for $x_a(t_f)$, $p_a(t_f)$, and t_{f_a}; note that $x_a(t_f)$ and $p_a(t_f)$ are Jacobean matrices of the form:

$$x_a(t_f) = \begin{bmatrix} x_{1a_1}(t_f) & \cdots & x_{1a_n}(t_f) \\ \vdots & & \vdots \\ x_{na_1}(t_f) & \cdots & x_{na_n}(t_f) \end{bmatrix} \quad . \quad (18)$$

$$p_a(t_f) = \begin{bmatrix} p_{1a_1}(t_f) & \cdots & p_{1a_n}(t_f) \\ \vdots & & \vdots \\ p_{na_1}(t_f) & \cdots & p_{na_n}(t_f) \end{bmatrix} \quad . \quad (19)$$

Assuming $Y = x_a$ and $Z = p_a$ it can be shown that (*see* Appendix 4.II)

$$\frac{dY}{dt} = f_x Y + f_p Z, \qquad Y(0) = 0 \quad . \quad (20)$$

and

$$\frac{dZ}{dt} = g_p Z + g_x Y, \qquad Z(0) = 1 \quad . \quad (21)$$

Equations (20) and (21) will be referred to as the accessory equations. For the particular case of bang bang control,

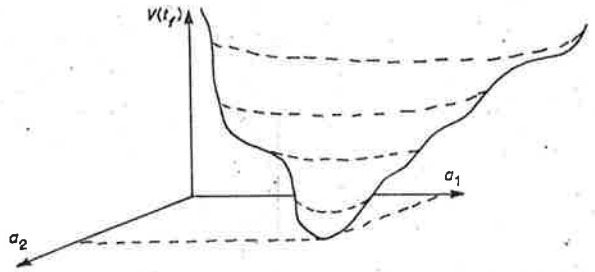


Fig. 4.2. The error criterion depicted as a 'valley'

equation (20) must be handled in a special manner but this presents no difficulties (*see* Appendix 4.III). The variables Y and Z may be considered as mapping functions: they describe the transformation of a at $t = t_0$ into x and p at $t = t_f$. Hence,

$$x_a(t_f) = Y(t_f) \quad . \quad . \quad . \quad . \quad (22)$$

and

$$p_a(t_f) = Z(t_f) \quad . \quad . \quad . \quad . \quad (23)$$

To determine t_{f_a}; differentiating equation (11) with respect to a yields

$$\frac{dW}{dt} = 0 \quad . \quad . \quad . \quad . \quad . \quad (24)$$

or

$$W_x x_a + \frac{dW}{dt} t_{f_a} + W_p p_a = 0 \quad . \quad . \quad (25)$$

But

$$\frac{dW}{dt} = W_x \frac{dx}{dt} + W_p \frac{dp}{dt} + W_{t_f} \quad . \quad . \quad (26)$$

Substituting equation (26) into equation (25) gives

$$t_{f_a} = \frac{-W_x x_a + W_p p_a}{W_x\, dx/dt + W_p\, dp/dt + W_{t_f}} \quad . \quad (27)$$

or using equations (22) and (23)

$$t_{f_a} = \frac{-W_x(t_f)Y(t_f) + W_p(t_f)Y(t_f)}{W_x(t_f)\, dx(t_f)/dt + W_p(t_f)\, dp(t_f)/dt + W_{t_f}(t_f)} \quad (28)$$

Substituting equations (22), (23), and (28) into equation (17), it is now possible, with a knowledge of x, p, t_f, Y, and Z, to calculate V_a.

The computational technique may be outlined with reference to the flow chart in Fig. 4.3:

(1) Assume a value of a and choose k.

(2) Integrate equations (8), (9), (20), and (21) until $W = 0$. Note that this involves the integration of $2(n^2+n)$ equations.

(3) Calculate the value of $V(t_f)$ in equation (14) and ascertain if $V = 0$. If it is, then the problem has been solved.

(4) If $V \neq 0$, check if the new value is smaller than the stored value of V. If not, halve the value of k and find a new value for a using the stored V_a. This is repeated until the new V determined falls below the value of the stored V.

(5) If step 4 has been satisfied, calculate $V_a(t_f)$ from equation (17) and update a by using equation (16).

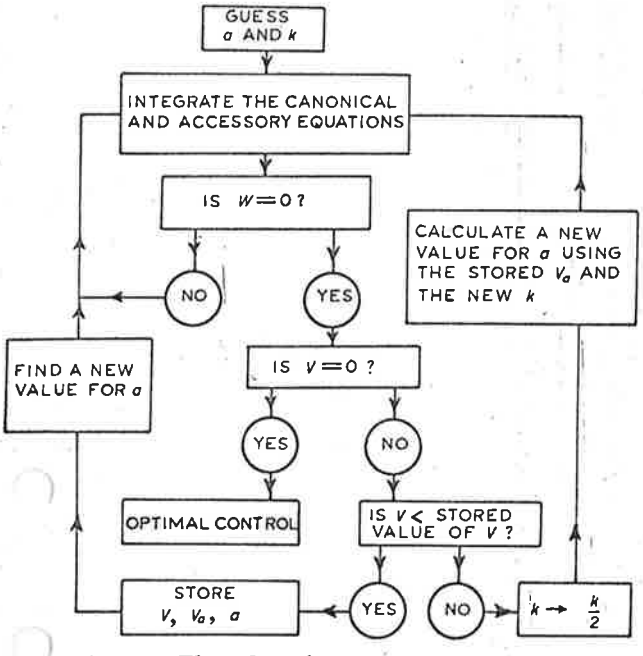


Fig. 4.3. Flow chart for computer programme

(6) Return to step 2 and repeat steps 2, 3, 4, 5, and 6 until $V = 0$.

Step 4 is necessary since it is not known what value of k to choose initially. Thus it seems best to commence with a 'large' value and reduce as required.

For time optimal problems it is possible to vary slightly the above procedure. Instead of choosing only the value of a for a minimum of V, it is possible to steepest descend by choosing a and t_f, the optimal time. This requires the value of the gradient of V with respect to t_f: thus

$$\frac{dV(t_f)}{dt_f} = V_x(t_f)\frac{dx(t_f)}{dt}+V_p(t_f)\frac{dp(t_f)}{dt}+V_{t_f}(t_f) \quad (29)$$

The time t_f is updated at each stage (along with a) by choosing

$$t_{f_{new}} = t_{f_{old}}-k\frac{dV(t_f)}{dt} \quad . \quad . \quad . \quad (30)$$

In this case an 'artificial' stopping condition must be chosen on the first run and

$$W = t-t_f \quad . \quad . \quad . \quad . \quad . \quad (31)$$

must then be used on all successive runs.

EXAMPLES

(i) *Harmonic oscillator*

Given a system described by the equations

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -0{\cdot}1x_1+u$$

it is required to find the control $u(t)$ which will transfer the system from the point $(-5, -1)$ to the origin in minimum time. An application of equations (56), (57), and (58) yields the adjoint equations

$$\frac{dp_1}{dt} = 0{\cdot}1p_2, \qquad p_1(0) = a_1$$

$$\frac{dp_2}{dt} = -p_1, \qquad p_2(0) = a_2$$

and the control

$$u(t) = \text{sgn}\,[p_2(t)]$$

Thus we have

$$f_1 = x_2$$
$$f_2 = -0{\cdot}1x_1+\text{sgn}\,(p_2)$$
$$g_1 = 0{\cdot}1p_2$$
$$g_2 = -p_1$$

Using equations (20) and (21), the accessory equations become

$$\frac{dY_{11}}{dt} = Y_{21} \qquad Y_{11}(0) = 0$$

$$\frac{dY_{12}}{dt} = Y_{22} \qquad Y_{12}(0) = 0$$

$$\frac{dY_{21}}{dt} = -0{\cdot}1\,Y_{11}+\frac{2(-1)^j\,\text{sgn}\,(a_2)\delta_{t,y_j}Z_{21}}{dp_2/dt} \qquad Y_{21}(0) = 0$$

$$\frac{dY_{22}}{dt} = -0{\cdot}1\,Y_{12}+\frac{2(-1)^j\,\text{sgn}\,(a_2)\delta_{t,y_j}Z_{22}}{dp_2/dt} \qquad Y_{22}(0) = 0$$

$$\frac{dZ_{11}}{dt} = 0{\cdot}1Z_{21} \qquad Z_{11}(0) = 1$$

$$\frac{dZ_{12}}{dt} = 0{\cdot}1Z_{22} \qquad Z_{12}(0) = 0$$

$$\frac{dZ_{21}}{dt} = -Z_{11} \qquad Z_{21}(0) = 0$$

$$\frac{dZ_{22}}{dt} = -Z_{12} \qquad Z_{22}(0) = 1$$

Note that in the computer programme, the expression $2(-1)^j\,\text{sgn}\,(a_2)\delta_{t,y_j}$ was approximated by $u(t)-u(t-h)$, where h is the sampling time of the integration procedure.

The error function was defined as

$$V = x_1(t_f)^2+x_2(t_f)^2$$

so that the following equations must be chosen

$$a_{1_{new}} = a_{1_{old}}-2k[x_1(t_f)Y_{11}(t_f)+x_2(t_f)Y_{21}(t_f)]$$
$$a_{2_{new}} = a_{2_{old}}-2k[x_1(t_f)Y_{12}(t_f)+x_2(t_f)Y_{22}(t_f)]$$
$$t_{f_{new}} = t_{f_{old}}-2k[x_1(t_f)f_1(t_f)+x_2(t_f)f_2(t_f)]$$

To begin the computation

$$a_1 = 0{\cdot}2$$
$$a_2 = 1$$
$$k = 10.$$

were chosen. It was also necessary to define an 'artificial' stopping condition for the first run:

$$W = t-(1+t_1)$$

where $t = t_1$ is defined by the equation

$$Y_{12}(t_1)-0{\cdot}01 = 0$$

The solution was found after 33 steps (about 30 min computation time on the Mercury) when $V < 0{\cdot}005$. In order to speed up the convergence when V changed little from iteration to iteration, it was found useful to slightly perturb a_1, a_2, and t_f arbitrarily in an attempt to get off the computation 'plateau'. Fig. 4.4 is a plot of the state variables along the optimal trajectory; Fig. 4.5 shows the adjoint variables and the desired optimal control function. The solution to the accessory equations for the optimal control are shown in Figs 4.6 and 4.7.

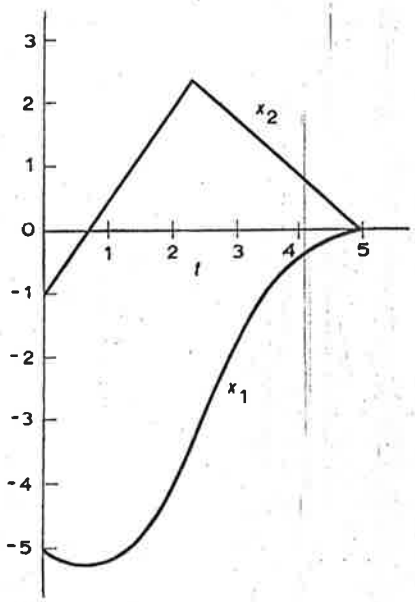


Fig. 4.4. The state variables

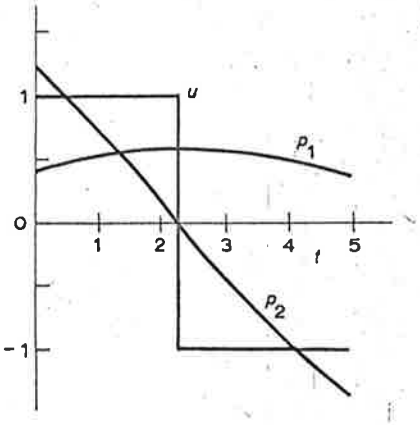


Fig. 4.5. The adjoint variables and the control function

(ii) *Rayleigh's equation*

The dynamic equations of the system are:

$$\frac{dx_1}{dt} = x_2 \qquad x_1(0) = -5$$

$$\frac{dx_2}{dt} = -x_1+1{\cdot}4x_2-0{\cdot}14x_2^3+4u \qquad x_2(0) = -5$$

it is required to minimize

$$P(u) = \int_0^{t_f} (x_1^2+u^2)\,dt$$

where $t_f = 100$ msec. Using equations (56), (57), and (58) yields the following adjoint equations:

$$\frac{dp_1}{dt} = p_2+2x_1$$

$$\frac{dp_2}{dt} = -p_1-1{\cdot}4p_2+0{\cdot}42p_2x_2^2, \quad p_2(0) = a_2$$

the control function is found to be

$$u(t) = 2p_2(t)$$

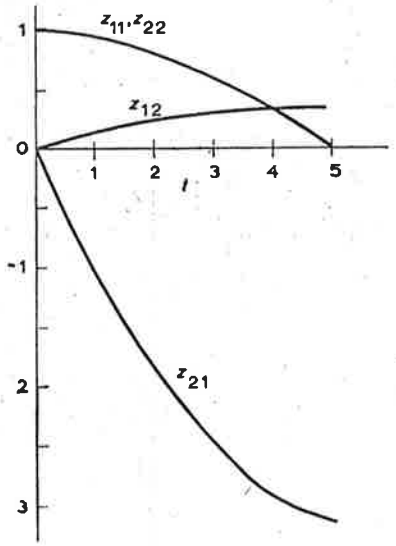


Fig. 4.6. The Z accessory extremals

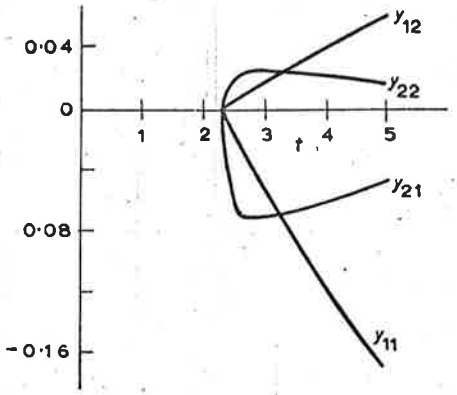


Fig. 4.7. The Y accessory extremals

t

An application of equations (20) and (21) yields the accessory equations,

$$\frac{dY_{11}}{dt} = Y_{22} \qquad Y_{11}(0) = 0$$

$$\frac{dY_{12}}{dt} = Y_{22} \qquad Y_{12}(0) = 0$$

$$\frac{dY_{21}}{dt} = -Y_{11}+1{\cdot}4Y_{21}-0{\cdot}42x_2^2Y_{21}+8Z_{21} \qquad Y_{21}(0) = 0$$

$$\frac{dY_{22}}{dt} = -Y_{12}+1{\cdot}4Y_{22}-0{\cdot}42x_2^2Y_{22}+8Z_{22} \qquad Y_{22}(0) = 0$$

$$\frac{dZ_{11}}{dt} = Z_{21}+2Y_{11} \qquad Z_{11}(0) = 1$$

$$\frac{dZ_{12}}{dt} = Z_{22}+2Y_{12} \qquad Z_{12}(0) = 0$$

$$\frac{dZ_{21}}{dt} = -Z_{11}-1{\cdot}4Z_{21}+0{\cdot}42x_2^2Z_{21} +0{\cdot}84p_2x_2Y_{21} \qquad Z_{21}(0) = 0$$

$$\frac{dZ_{22}}{dt} = -Z_{12}-1{\cdot}4Z_{22}+0{\cdot}42x_2^2Z_{22} +0{\cdot}84p_2x_2Y_{22} \qquad Z_{22}(0) = 1$$

The function V may be defined as

$$V = p_1^2+p_2^2$$

so that we must choose

$$a_{1_{new}} = a_{1_{old}}-2k[p_1(t_f)Z_{11}(t_f)+p_2(t_f)Z_{21}(t_f)]$$

$$a_{2_{new}} = a_{2_{old}}-2k[p_1(t_f)Z_{12}(t_f)+p_2(t_f)Z_{22}(t_f)]$$

Initially, $a_1 = a_2 = 0{\cdot}2$ and $k = 1{\cdot}0$. After 19 computational runs (about 20 min on the Mercury) $V < 10^{-6}$. The optimal values of the initial conditions for the adjoint equation were found to be $a_1 = 1{\cdot}05495$ and $a_2 = 0{\cdot}04266$. The control function is plotted in Fig. 4.8.

CONCLUSIONS

A method for finding the solution of the two-point boundary value problem which arises in the optimal control of systems has been described. The main advantage of this method is its inherent simplicity and low storage and time requirements on the digital computer.

In addition to the work reported here, the important control problem with state inequality constraints is being considered. Calculations are also being performed which demonstrate that by using a sensitivity matrix derived from the solution of the accessory equations, it is possible to synthesize a linear or non-linear feed-back controller which could operate in the region of an optimal trajectory.

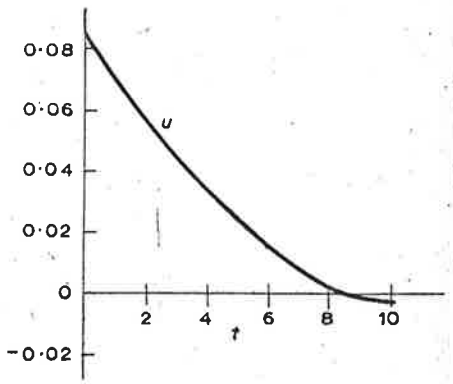


Fig. 4.8. The control function

ACKNOWLEDGEMENTS

The author would like to thank the Minister of Youth, Province of Quebec, and the National Research Council, Ottawa, Canada, for the financial support which enabled him to carry out this research.

The computations in this paper were performed on the Mercury Computer at the University of London Computer Unit.

APPENDIX 4.I

THE NECESSARY CONDITIONS FOR AN OPTIMAL CONTROL

With respect to the formulation of the control problem as described under the heading 'The control problem', it is required to examine the necessary conditions for an optimal control in respect of the Theory of the Calculus of Variations. The trajectory in phase space which satisfies these conditions will be referred to as an optimal trajectory, E^*.

To handle the control constraints, it is necessary to define a 'slack' variable, $c_i(t)$, $i = 1,\ldots,k$ such that

$$C_i(x, u, t)-c_i^2 = 0 \quad . \quad . \quad . \quad . \quad . \quad (32)$$

the magnitude of $c(t)$ is chosen continuously to arbitrarily construct this equality constraint from the original inequality constraint, equation (2). Thus in addition to the n dynamic constraints of equation (1), the k equations of equation (32) must also be satisfied.

The problem may be solved by utilizing the Multiplier Rule (3) and the pertinent corollaries:

(i) For an optimal trajectory, there exists a constant $p_0 < 0$, an n-dimensional vector multiplier $p(t)$ such that $p \in \bar{P}$, a k-dimensional vector multiplier $e(t)$ such that

$$\begin{bmatrix} p_0(t) \\ \cdots \\ p(t) \\ \cdots \\ e(t) \end{bmatrix} \neq 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad (33)$$

the multipliers are continuous except possibly at corners of E^* where they have unique and well defined right and left limits. In addition, there exists a scalar function

$$F\left(t, x, c, \frac{dx}{dt}, u, p_0, p, e\right) = p_0f_0+p\left(f-\frac{dx}{dt}\right)+e(C-c^2) \quad . \quad (34)$$

which must satisfy the following Euler–Lagrange equations:

$$\frac{dF_{x'}}{dt} = F_x \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (35)$$

$$F_u = \text{constant} \quad . \quad . \quad . \quad . \quad . \quad . \quad (36)$$

$$F_c = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (37)$$

($x' = dx/dt$). Equation (35) may be considered as the basic Euler–Lagrange equation if x is treated as a dummy variable. Note that since u is actually a control on the time derivative of x,

it must be handled in the same manner as dx/dt. As a consequence of this, if

$$v = \int_{t_0}^{t} u\,dt, \quad (dv/dt = u) \quad . \quad . \quad . \quad (38)$$

then an application of equation (35) yields

$$F_v = 0 \quad . \quad . \quad . \quad . \quad . \quad (39)$$

and equation (36) results. On the other hand, c is treated in the same way as x, and since dc/dt does not appear in equation (34), then $F_{c'} = 0$ and equation (39) results.

(ii) At the termination of E^*, in addition to the p equations of equation (7), the following transversality condition must also hold:

$$\left[\left(F-\frac{dx}{dt}F_{x'}\right)dt+F_{x'}\,dx+F_u\,dv\right]_{t=t_f} = 0 \quad . \quad (40)$$

since v is not fixed at $t = t_f$, F_u must vanish. Equations (36) and (40) may now be rewritten as

$$F_u = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad (41)$$

$$\left[\left(F-\frac{dx}{dt}F_{x'}\right)dt+F_{x'}\,dx\right]_{t=t_f} = 0 \quad . \quad . \quad (42)$$

respectively.

(iii) In addition to the basic continuity requirements stated under the heading 'The control problem', the Weierstrass–Erdmann corner condition also holds at a discontinuity $t = \gamma_j$, $j = 1, 2, \ldots$, the functions $F_{x'}$, F_u, and $[F-(dx/dt)F_{x'}]$ are continuous or

$$F_{x'}{}^{-}(\gamma_j-0) = F_{x'}{}^{+}(\gamma_j+0) \quad . \quad . \quad . \quad (43)$$

$$F_u{}^{-}(\gamma_j-0) = F_u{}^{+}(\gamma_j+0) \quad . \quad . \quad . \quad (44)$$

$$F^{-}(\gamma_j-0)-\frac{dx(\gamma_j-0)}{dt}F_{x'}{}^{-}(\gamma_j-0) = F^{+}(\gamma_j+0)-\frac{dx(\gamma_j+0)}{dt}F_{x'}{}^{+}(\gamma_j+0) \quad (45)$$

Note that for many problems, points (i), (ii), and (iii) plus physical reasoning are sufficient to ensure that in fact the trajectory is optimal.

(iv) Every normal (3) maximizing trajectory E^* must satisfy the necessary conditions of Weierstrass: For every element $(t, x, c, dx/dt, u, p_0, p, e)$ of E^* and for all admissible sets $(t, x, c, \overline{dx}/dt, \bar{u}) \neq (t, x, c, dx/dt, u)$ which satisfy equation (1),

$$E\left(t, x, c, \frac{dx}{dt}, u, p_0, p, e\right) \leqslant 0 \quad . \quad . \quad . \quad (46)$$

where

$$E = F\left(t, x, c, \frac{\overline{dx}}{dt}, \bar{u}, p_0, p, e\right)-F\left(t, x, c, \frac{dx}{dt}, u, p_0, p, e\right)-\left(\frac{\overline{dx}}{dt}-\frac{dx}{dt}\right)L_{x'}-(\bar{u}-u)L_u \quad (47)$$

An application of equation (35) to equation (34) yields

$$\frac{dp}{dt} = -p_0 f_{0_x}-pf_x-eC_x \quad . \quad . \quad . \quad . \quad (48)$$

these are the so-called adjoint equations. From equations (35) and (37) it can be seen that

$$F_{c_i} = -2e_i c_i = 0 \quad . \quad . \quad . \quad . \quad (49)$$

Multiplying both sides of equation (32) by e_i yields

$$e_i C_i = 0, \quad i = 1, \ldots, k \quad . \quad . \quad . \quad (50)$$

Thus it may be deduced that off a boundary of U, $e_i = 0$, while on a boundary, $C_i = 0$.

Equations (36) and (34) give

$$p_0 f_{0_u}+pf_u+eC_u = 0 \quad . \quad . \quad . \quad . \quad (51)$$

The k equations of equation (50) and the r equations of equation (51) are sufficient to determine the $k+r$ variables e and u.

Applying the Weierstrass–Erdmann corner condition of equation (43) to equation (34) yields

$$p^{-}(\gamma_j-0) = p^{+}(\gamma_j+0) \quad . \quad . \quad . \quad (52)$$

or equivalently p is continuous in t. Similarly, from equations (46) and (47) it is found that the expressions $(p_0 f_{0_u}+pf_u+eC_u)$ and $(p_0 f_0+pf)$ are continuous.

The transversality condition of equation (42) yields

$$[(p_0 f_0+pf)\,dt-p\,dx]_{t=t_f} = 0 \quad . \quad . \quad . \quad (53)$$

It is convenient to express these results in the form of Pontryagin's Maximum Principle (8). For a *minimum* of $P(u)$ we must now set $p_0 = -1$; also let

$$p = \begin{bmatrix} -1 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} \quad . \quad (54)$$

and define a Hamiltonian function

$$H = pf \quad . \quad . \quad . \quad . \quad . \quad . \quad (55)$$

Then the conditions for optimality become

$$\frac{dx}{dt} = f \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (56)$$

$$\frac{dp}{dt} = -H_x-eC_x \quad . \quad . \quad . \quad . \quad . \quad (57)$$

$$H_u+eC_u = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad (58)$$

and

$$[H\,dt-p\,dx]_{t=t_f} = 0 \quad . \quad . \quad . \quad . \quad (59)$$

Lastly, the Weierstrass condition yields for every (t, x^*, u^*, p) where x^* and u^* are optimal functions and for any $u \in U$,

$$H(t, x^*, u, p) \leqslant H(t, x^*, u^*, p) \quad . \quad . \quad . \quad (60)$$

Note that H is continuous. Also x and p are called the canonical variables and equations (56) and (57) are the canonical equations.

Using equations (58), (60), and (50), it is possible to eliminate u and e from equations (56) and (57); the equations (8) and (9) are thus obtained.

APPENDIX 4.II

THE ACCESSORY EQUATIONS

From the analysis of Appendix 4.I the optimization problem of choosing a control function when the latter is constrained, was reduced to finding a solution to the two-point boundary value problem. In the computational scheme described under the heading 'The computational technique', it was found necessary to evaluate x_a and p_a as continuous functions of time. By treating a as a parameter in the functions f and g of equations (8) and (9), we may examine the differentiability properties of the trajectory $x[t; t_0, x(t_0); p, a]$ and $p(t; t_0, a; x)$. The former will be considered first. It will be required to prove that the solution of the state equations (8) is of class C^1 in a.

Basically, the proof depends on Gronwall's Lemma (21) (22) (23) (24). Because this lemma is not commonly familiar, a proof will be given which follows closely Lefschetz (24).

GRONWALL'S LEMMA: Let $f(t)$ be a vector function such that

$$0 \leqslant f(t) \leqslant b_1+\int_{t_0}^{t}[B_2 f(s)+b_3]\,ds \quad . \quad . \quad (61)$$

where b_1 and b_3 are n-dimensional positive vector constants and B_2 is an $n \times n$ constant matrix with all its elements positive. The function $f(t)$ is continuous in the range $t_0 \leqslant t \leqslant t_f$. If $t_1-t_f = \rho$, then

$$f(t) \leqslant (b_3\rho+b_1)\,e^{B_2\rho}, \quad t_0 \leqslant t \leqslant t_f \quad . \quad . \quad (62)$$

Proof: Let

$$f(t) = b(t)\,e^{B_2(t-t_0)} \quad . \quad . \quad . \quad . \quad . \quad (63)$$

so that $b(t)$ is continuous for $t_0 \leqslant t \leqslant t_f$. If

$$b_m = \sup b(t) \quad . \quad . \quad . \quad . \quad . \quad (64)$$

in this interval of time, $b(t)$ has the value b_m for some $t = t_m$. Equation (61) gives

$$\begin{aligned} b_m\,e^{B_2(t_m-t_0)} &\leqslant b_1 + \int_{t_0}^{t_m} (B_2 b_m\,e^{B_2(s-t_0)} + b_3)\,ds \\ &= b_1 + b_3(t_m - t_0) + b_m(e^{B_2(t_m-t_0)} - 1) \end{aligned} \quad (65)$$

or

$$b_m \leqslant b_1 + b_3$$

Thus

$$f(t) \leqslant (b_3\rho + b_1)\,e^{B_2\rho}, \quad t_0 \leqslant t \leqslant t_f$$

and the lemma is proved.

Now suppose that

$$f(t) = b_1 + \int_{t_0}^{t} [B_2 f(s) + b_3]\,ds \quad . \quad . \quad . \quad (66)$$

Using the Schwartz inequality it can be shown that

$$0 \leqslant |f(t)| \leqslant |b_1| + \int_{t_0}^{t} [|B_2| \,.\, |f(s)| + |b_3|]\,ds \quad . \quad (67)$$

A direct application of the lemma yields

$$|f(t)| \leqslant (|b_3|\rho + |b_1|)\,e^{|B_2|\rho} \quad . \quad . \quad . \quad (68)$$

It can be shown that if $f(x, p, t)$ is piecewise continuous in $X \times \bar{P}$ and bounded in $X \times \bar{P} \times T$, and satisfies a Lipschitz condition, then the solution of the dynamic equations $x[t; t_0, x(t_0); p, a]$ is continuous. Another important property of the trajectory is uniqueness: only one solution of equation (8) can pass through a given point in $X \times \bar{P} \times T$ (**25**). In order to demonstrate that x is of class C^1 it is necessary to show that the function x_a exists and is continuous in a. It is assumed that the time interval involved is finite so that $|t - t_0| \leqslant \rho$.

Equation (8) and the corresponding initial conditions are equivalent to

$$x(t) = x(t_0) + \int_{t_0}^{t} f(x, p, s)\,ds \quad . \quad . \quad . \quad (69)$$

Using a Taylor series expansion of equation (69) and dividing both sides of the equation by Δa yields

$$\frac{\Delta x}{\Delta a} = \int_{t_0}^{t} \left(f_x \frac{\Delta x}{\Delta a} + f_p \frac{\Delta p}{\Delta a} + \delta_1 \right) ds \quad . \quad . \quad . \quad (70)$$

where δ_1 is a remainder term such that $\delta_1 \to 0$ as $a \to 0$ uniformly in t for $|t - t_0| \leqslant \rho$. To proceed further it is necessary to examine the following differential equation:

$$\frac{dY}{dt} = f_x Y + f_p p_a, \quad Y(t_0) = 0 \quad . \quad . \quad . \quad (71)$$

This matrix differential equation is linear in Y, and because of the assumptions on f, it has a solution which is unique and continuous in t. The solution of equation (71) may be represented as

$$Y(t) = \int_{t_0}^{t} (f_x Y + f_p p_a)\,ds \quad . \quad . \quad . \quad (72)$$

Subtracting equation (72) from equation (70) and defining $\sigma_1 = \Delta x/\Delta a - Y$, gives

$$\sigma_1(t) = \int_{t_0}^{t} (f_x \sigma_1 + \delta_1)\,ds \quad . \quad . \quad . \quad (73)$$

Owing to the continuity assumptions on f, it may be stated that f_x has an upper bound B_2 in $|t - t_0| \leqslant \rho$. Applying equation (68) directly yields

$$|\sigma_1(t)| \leqslant \delta_1 \rho\,e^{B_2\rho} \quad . \quad . \quad . \quad . \quad . \quad (74)$$

Since $\delta_1 \to 0$ as $\Delta a \to 0$, it is obvious that $|\sigma_1(t)| \to 0$ as $\Delta a \to 0$. Thus $x_a \to Y$ as $\Delta a \to 0$ and x_a exists and is continuous in t in $|t - t_0| \leqslant \rho$. The solution of equation (71) yields x_a as a function of time. The extension to class C^r is obvious and so the theorem is proved.

If $Z = p_a$, equation (42) becomes

$$\frac{dy}{dt} = f_x Y + f_p Z, \quad Y(t_0) = 0 \quad . \quad . \quad . \quad (75)$$

It still remains to examine the differentiability properties of equation (9). Analogous reasoning as before yields

$$p_a = I + \int_{t_0}^{t} \left(g_p \frac{\Delta p}{\Delta a} + g_x \frac{\Delta x}{\Delta a} + \delta_2 \right) ds \quad . \quad (76)$$

where a is treated as a parameter. Also

$$\frac{dZ}{dt} = g_p Z + g_x x_a, \quad Z(t_0) = I \quad . \quad . \quad . \quad (77)$$

or

$$Z(t) = I + \int_{t_0}^{t} (g_p Z + g_x x_a)\,ds \quad . \quad . \quad . \quad (78)$$

may be considered. Again

$$|\sigma_2| = \left| \frac{\Delta p}{\Delta a} - Z \right| \leqslant \delta_2 \rho\,e^{B_2\rho} \quad . \quad . \quad . \quad (79)$$

and $\sigma_2 \to 0$ as $\Delta a \to 0$. Thus equation (77) may be used to determine p_a as a continuous function of time. Equation (77) may be rewritten as

$$\frac{dZ}{dt} = g_p Z + g_x Y, \quad Z(t_0) = I \quad . \quad . \quad . \quad (80)$$

Equations (71) and (80) are called the accessory equations. It is interesting to note that the latter are equivalent to the Jacobi differential equations which are the extremals arising out of the problem of extremizing the second variation of $P(u)$ (**3**). Thus, the solutions of these equations are called the accessory extremals.

APPENDIX 4.III

THE EFFECT OF A BANG BANG CONTROL FUNCTION ON THE ACCESSORY EQUATIONS

For many problems the control which arises out of the optimization procedure is piecewise continuous or, equivalently, bang bang. Thus, as an example,

$$u(t) = \text{sgn}\,[R(p)] \quad . \quad . \quad . \quad . \quad . \quad (81)$$

may occur as the control; u is plotted as a function of R in Fig. 4.9. This u will be used to demonstrate the technique. Some difficulty results as far as the functions f_p and g_p in equations (20) and (21) are concerned. The following discussion holds for both functions although only the former will be considered.

Writing

$$f_p = f_u u_R R_p \quad . \quad . \quad . \quad . \quad . \quad (82)$$

there is no difficulty in finding f_u and R_p directly by differentiation; the problem involves u_R. Equation (20) may be rewritten as

$$Y(t) = \int_{t_0}^{t_f} (f_x Y + f_p Z)\,ds \quad . \quad . \quad . \quad (83)$$

however, being interested only in the second term of the integrand and using equation (82) in equation (83) an integral I is considered such that

$$I = \int_{t_0}^{t_f} (f_u u_R R_p Z)\,ds \quad . \quad . \quad . \quad . \quad (84)$$

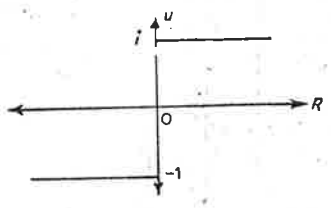


Fig. 4.9. The variable u as a function of R

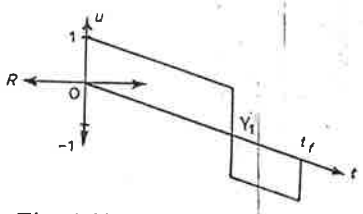


Fig. 4.10. The control function

Let γ_1 be a switching point in the time interval (t_0, t_f). The extension to the case of more than one switchover will be obvious. Fig. 4.10 shows the variable u as a function of time t.

Using equations (81) and (84)

$$I = \int_{t_0}^{t_f} (f_u) \left(\lim_{\delta R \to 0} \frac{\operatorname{sgn}\left(R+\frac{\delta R}{2}\right) - \operatorname{sgn}\left(R-\frac{\delta R}{2}\right)}{\delta R} \right) \delta R \, \frac{ds}{dR} \, (R_p Z) \, ds \quad . \quad . \quad . \quad (85)$$

$$= (f_u) \left[2(-1) \operatorname{sgn} [R(t_0)] \delta_{t,\gamma_1} \left(\frac{dR(\gamma_1)}{dt} \right)^{-1} \right] (R_p Z) \quad . \quad . \quad . \quad (86)$$

where δ_{t,γ_1} is the Kronecker delta function. If switching occurs at $\gamma_j, j = 1, 2, \ldots$, then

$$I = \sum_j (f_u) \left(2(-1)^j \operatorname{sgn} [R(t_0)] \delta_{t,\gamma_j} \frac{dR(\gamma_j)^{-1}}{dt} \right) [R_p(\gamma_j) Z(\gamma_j)] \quad (87)$$

Substituting equation (87) into equation (83) yields

$$Y(t) = \int_{t_0}^{t_f} (f_x Y) \, ds + I \quad . \quad . \quad . \quad (88)$$

APPENDIX 4.IV

REFERENCES

(1) EYKHOFF, P. 'Some fundamental aspects of process-parameter estimation', *I.R.E. Trans. auto. Control* 1963 **AC-8** (Oct.), 347.

(2) BLISS, G. A. 'The problem of Lagrange in the calculus of variations', *Amer. J. Math.* 1930 **L II**, 673.

(3) BLISS, G. A. 'Lectures on the calculus of variations', Phoenix Science Series, 1961 (University of Chicago Press).

(4) BOLZA, O. *Lectures on the calculus of variations* 1961 (Dover Publications, New York).

(5) PEARSON, J. D. 'Studies in the optimal control of dynamic systems', Ph.D. Thesis, August 1963 (University of London).

(6) BERKOVITZ, L. D. 'Variational methods in problems of control and programming', *J. math. Analysis Applic.* 1961 **3** (Aug., No. 1).

(7) HESTENES, M. R. 'A general problem in the calculus of variations with applications to paths of least time', Rand Corporation *Research Memorandum RM-100*, 1950 (March).

(8) PONTRYAGIN, L. S., BOTTYANSKII, V. G., GAMKRELIDZE, R. V. and MISCHENKO, E. F. *The mathematical theory of optimal processes*, English trans., K. N. Trirogoff, 1962 (Interscience Publishers, New York).

(9) KELLEY, H. J., KOPP, R. E. and MAYER, H. 'Successive approximation techniques for trajectory optimization', Proc. I.A.S. Symposium on Vehicle Systems Optimization, New York, November 1961.

(10) BRYSON, A. E. Jr. and DENHAM, W. F. 'A steepest-ascent method for solving optimum programming problems', *J. appl. Mech.* 1962 **29** (Series E, No. 2), 247.

(11) DREYFUS, S. 'Variational problems with state variable inequality constraints', Rand Corporation *Report No. p-2605*, July 1962.

(12) BRYSON, A. E. Jr. and DENHAM, W. F. 'The solution of optimal programming problems with inequality constraints', Raytheon *Report BR-2121*, November 1962.

(13) KELLEY, H. J. *Optimization techniques*, Chap. 6, 'Method of gradients', Ed. G. Leitmann, 1962 (Academic Press, New York).

(14) KIPINIAK, W. *Dynamic optimization and control* 1961 (M.I.T. Press and John Wiley and Sons, Inc.).

(15) PAIEWONSKY, B. 'A study of time optimal control', A.R.A.P. *Report No. 33*, July 1961.

(16) NEUSTADT, L. W. 'Synthesizing time optimal control systems', *J. math. Analysis Applic.* 1960 **1**, 484.

(17) BREAKWELL, J. V. 'The optimization of trajectories', *J. Soc. ind. appl. Math.* 1959 **7**.

(18) JUROVICS, S. A. and MCINTYRE, J. E. 'The adjoint method and its application to trajectory optimization', *ARS Jl.* 1962 **32** (Sept., No. 9).

(19) KULAKOWSKI, L. J. and STANCIL, R. T. 'Rocket boost trajectories for maximum burn-out velocity', *ARS Jl.* 1960 **30** (July, No. 7).

(20) RAJARAMAN, V. and WERTZ, J. 'On stability and steepest descent', *I.R.E. Trans. auto. Control* 1963 **AC-8** (Jan., No. 1).

(21) GRONWALL, T. H. 'Note on the derivatives with respect to a parameter of the solutions of differential equations', *Ann. Math.* 1919 **20** (Series 2), 292.

(22) RITT, J. F. 'On the differentiability of the solution of a differential equation with respect to a parameter', *Ann. Math.* 1919 **20** (Series 2), 289.

(23) CODDINGTON, E. A. and LEVINSON, N. *Theory of ordinary differential equations* 1955 (McGraw-Hill, New York).

(24) LEFSCHETZ, S. *Differential equations: Geometric theory*, 2nd ed. 1957 (Interscience Publishers, New York).

(25) BLISS, G. A. 'The solution of differential equations of the first order as functions of their initial values', *Ann. Math.* 1904–5 **6** (Series 2).