



自脱コンバインの最適走行制御

鬼頭, 孝治

(Degree)

博士 (学術)

(Date of Degree)

1985-10-15

(Date of Publication)

2008-04-25

(Resource Type)

doctoral thesis

(Report Number)

甲0559

(URL)

<https://hdl.handle.net/20.500.14094/D1000559>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

自脱コンバインの最適走行制御

昭和60年8月

神戸大学大学院自然科学研究科

鬼頭孝治

目 次

第 I 章	緒 言	1
第 II 章	稲調査	4
第 III 章	実験装置及び制御方法	7
III - 1	供試コンバイン	7
III - 2	各種制御装置	10
III - 2 - (a)	操向制御装置	10
III - 2 - (b)	刈取り部昇降制御装置	11
III - 2 - (c)	パワーシフト制御装置	12
III - 2 - (d)	油圧装置	13
III - 3	電気制御装置	14
III - 3 - (a)	メインマイクロコンピュータ	15
III - 3 - (b)	入出インターフェイス (メイン CPU 部)	16
III - 3 - (c)	サブマイクロコンピュータ	18
III - 3 - (d)	入出インターフェイス (サブ CPU 部)	18
III - 3 - (e)	A / D 変換器	19
III - 3 - (f)	電源供給回路	21
III - 4	旋回角検出方法	22
III - 4 - (a)	角速度センサ	22
III - 4 - (b)	角速度センサの性能試験	24
III - 4 - (c)	試験結果及び考察	25
III - 4 - (d)	旋回角演算理論	26
III - 4 - (e)	制御プログラム	29
III - 4 - (f)	旋回角演算装置の性能試験	31
III - 4 - (g)	試験結果及び考察	31
III - 5	メイン CPU 部の制御プログラム	33

Ⅲ-5-(a)	通常自動走行プログラム	38
Ⅲ-6	旋回実験	40
Ⅲ-6-(a)	実験結果及び考察	40
Ⅲ-7	旋回角検出装置の理論的解析	44
Ⅲ-8	旋回実験のシミュレーション	47
Ⅲ-8-(a)	供試コンパインの振動解析	47
Ⅲ-8-(b)	コンパインの走行モデル	50
Ⅲ-8-(c)	機体移動の概念	51
Ⅲ-8-(d)	旋回シミュレーション実験	53
Ⅲ-8-(e)	実験結果及び考察	54
第 IV 章	回行制御	58
IV-1	回行方法	58
IV-2	制御プログラム	60
IV-3	実験方法	61
IV-4	実験結果及び考察	61
第 V 章	最短距離制御	68
V-1	最短距離制御の概念	68
V-2	制御方法	69
V-2-(a)	稲列データの収集方法	69
V-2-(b)	制御プログラム	70
V-2-(c)	最短距離演算方法	71
V-2-(d)	制御プログラム	73
V-3	実験方法	73
V-4	実験結果及び考察	75
摘 要		81
謝 辞		86
参考文献		87
付 録		92

第 I 章 緒 言

農業機械の普及は、我が国農業の基幹作物である水稲作を中心に著しい進展をしてきた。昭和30年代には、歩行型トラクタ、動力噴霧機、動力散粉機を中心とした小型機械の開発改良と普及が急速に進み、40年代に入ってから、永年稲作栽培において機械化のネックとされた田植え及び収穫作業について、我が国独自の方式による田植機と自脱コンバイン等が開発され、乗用トラクタの改良・普及とあいまって、稲作の機械化一貫作業体系が完成した。現在は、更にこれらの機械の普及の進展と併せて、大型化・高性能化の質的發展をみせ、重労働からの解放、生産性の向上に大きな役割を果たしてきた。農業機械の普及状況は「昭和58年農業調査結果概要」（農林水産省統計情報部、58年1月時点）によれば58年の普及台数はトラクタ 440万台、自脱コンバイン 101万台余りとなっている。しかしながら、最近の農業事情により兼業農家の増加に伴い、特別な知識や技術を必要とせず、安全かつ高能率な農作業が可能で、かつメンテナンスが容易な上に故障も少ないことを不可欠条件とした機械が望まれている。

このような背景から、近年、農業機械の自動化が盛んに推し進められており、特にトラクタ、コンバインにおいてその傾向が顕著である。トラクタではプラウ、ロータリ等の耕深制御¹⁻⁹⁾、傾斜地作業での姿勢制御¹⁰⁻¹²⁾、各種誘導方式による自動操縦¹³⁻³⁰⁾、また最近では作業機を前後に装着し、赤外線センサによって検出された未耕地と既耕地の境界を目標経路として追従し、自動耕うんを行う研究³¹⁻³⁴⁾も報告されている。

一方、自脱コンバインは刈取り、脱穀、選別及びワラ処理等の多作業を同時に行う自走式のプロセッシングマシンと言えるが、これらの作業は機械の走行速度、穀物の生育状態及び土壌条件等により影響を受けるため、オペレータは各作業部の調節を頻繁に強いられ作業負担が増大する。これらを少しでも軽減するために、稲列に沿って自動的に追従する操向制御³⁵⁻³⁸⁾、圃場面の起伏に応じて刈取り部を常に最適な高さにする刈高さ制御³⁹⁻⁴⁴⁾、脱穀部供給口にお

いて、こぎ胴に対する穂先の相対位置を検出し常に最適な位置を保つこぎ深さ制御、脱穀部に供給される稲の量を検出し脱穀部の負荷を一定にする供給量制御⁴⁵⁻⁴⁹⁾等の研究例がある。また圃場末端部での回行及び各種警報装置等を含めたシステムの自動制御の研究⁵⁰⁻⁵³⁾も報告されている。

以上のように既に回行制御を含めたシステムの自動制御も報告されているが、本研究では更に一歩進めて、特に移動機械としてのコンバインに注目して回行制御と刈取り作業時の能率向上を目的として、より高精度で高安定なシステムを目指して研究を行った。自脱コンバインの回行の自動化は制御対象である稲と自脱コンバインとの相対位置を検出し、フィードバック制御を行うことにより可能となるが、現実には回行時に稲から離れるために相対位置の検出は非常に困難となる。本研究では稲と自脱コンバインとの相対位置を検出するのではなく、自脱コンバインの旋回角を検出し、マイクロコンピュータによるプログラム制御とで回行シーケンスを行った。旋回角の検出は磁気コンパスを利用したセンサ^{54, 55)}が振動及び傾斜による誤差が大であること、及び周囲の磁界の影響を受けること等の欠点を有するため、より高精度で分解能の高い角速度センサに換えて行った。これにより絶対的な旋回角が検出可能となり実旋回角をフィードバックして制御を行うことにより走行部でのスリップによる影響を減らすことが可能となった。

最近、田植機による移植が盛んに行われ植付け条数も複数条化しており、オペレータの熟練度にもよるが一般的に稲列の曲がりを伴うことが多い。詳細は第Ⅱ章に譲るが、稲の曲がりに関する調査の結果、約40m間に25cm p-pの振幅を持つことが判明した。また地上高8cmにおける株の直径は標本数677の平均で約7cmであった。本研究では2条刈りの自脱コンバインを供試するが、この刈幅は約77cmであり、稲の条間を30cmとすれば株間は37cmとなり40cmの刈取り余裕があると言える。これは目標を適切に選択し、稲列の曲がりの最大振幅が20cm p-p/条以下であれば、全く操舵することなく、直進走行が可能であり能率向上に役立つと言える。しかしながら、目標を適切に選択することは不可能に近く現実的ではない。そこで2条刈りのコンバインと複数条植え（偶数条）の田

植機に注目し、最初の刈取り作業で稲の曲がりのデータを収集し、2回目以降はそのデータを基に最短距離を計算して、刈取り作業を実施する制御をしようとするものであり、多条植えになるほど有効である。但し、横刈り走行では稲列が形成されていないためデータの収集は行わない。この制御を最短距離制御と呼ぶ。このような研究目的で自脱コンバインの自動制御システムの開発を行い、その可能性について検討した。

第 II 章 稲調査

自動制御のシステム設計に先立ち、制御対象となる稲の状態を把握することが重要となるため、昭和56年11月京都府小椋池において稲の生育状態及び稲列の曲がり方を調査した。品種は日本晴で、稲の地上高 8cmにおける株の直径、地上からの伸長時の草丈を標本数 677について測定した。また稲列の曲がり方は株跡約45m間にわたり基準線を設定し株毎（平均約18.5cm）の変位を測定した。それぞれの基礎統計量を表2-1にを示す。

表 2-1 稲調査結果 (品種：日本晴) 単位：c m

項 目	標本数	最大値	最小値	平均	標準偏差
株 径	677	10.0	3.0	6.6	0.99
株 丈	677	122.5	90.0	108.0	4.37
稲列変位	242	10.0	-15.0	-1.8	6.04

株径の変動範囲は3.0～10.0cmであり変位量は7cmである。平均値±2cm以内に約95%が含まれている。図2-1に度数分布を示す。これは双峯の正規分布を表しているが母集団の分布が双峯とは考えにくく、この標本特有の分布と考えるべきである。これにより刈幅に対する稲の占める割合が明確になった。

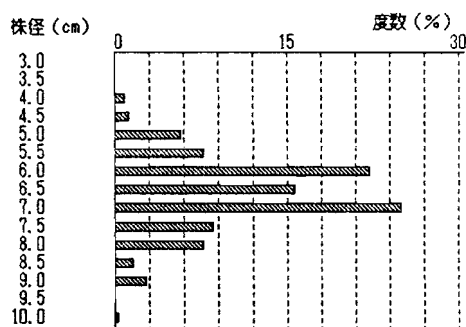


図 2-1 株径度数分布

草丈の変動範囲は90.0～122.5cmであり変位量は 32.5cmである。平均値±9cm以内に約95%が含まれている。図2-2に度数分布を示す。ほぼ正規分布しており、これは刈高さを一定とした場合、こぎ深さの調節範囲が20cm以上必要であると言える。又、変動の周期性を調べるために自己相関関数と

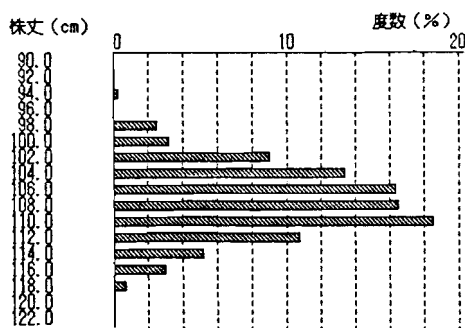


図 2-2 草丈度数分布

短いデータからも分解能の高い安定なスペクトルが得られる最大エントロピー法 (MEM)⁵⁶⁾ によって周波数分析を行った。図2-3にオートコログラム、図2-4にパワースペクトルを示す。このデータは3つの調査区に分けて収集したため連続ではなく、解析はそれぞれ別々に行った。なおそれぞれのデータ数はほぼ等分されており、一つの調査区は約40mである。横軸のラグ及び周波数は株間18.5cm、走行速度0.5m/sとして換算し、パワースペクトルは最大値を100として表示した。3つの調査区とも非常に似た傾向があり、数少ない標本の中でさえ、一定の周期性を見ることができる。最大周波数は0.01~0.02Hz付近、次に0.2~0.5Hz間に3つのピークが見られる。またR1の最大周波数が0Hzとなっているが、これはデータ不足のため非常に長い周期成分の存在を検出できなかったためと考えられる。これによってこぎ深さ制御装置の周波数応答が0.5Hz以上必要なことがわかる。

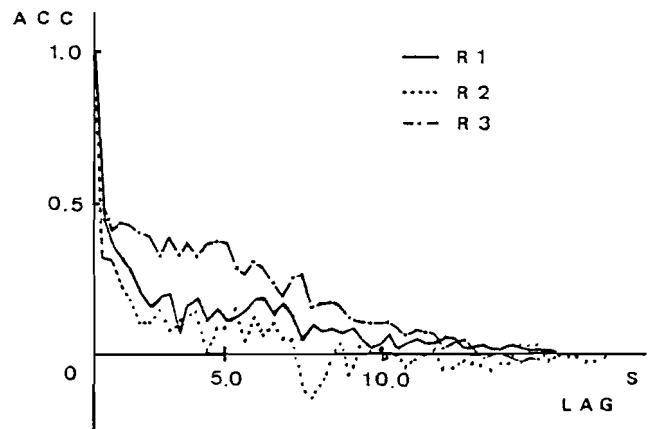


図 2-3 草丈のオートコログラム

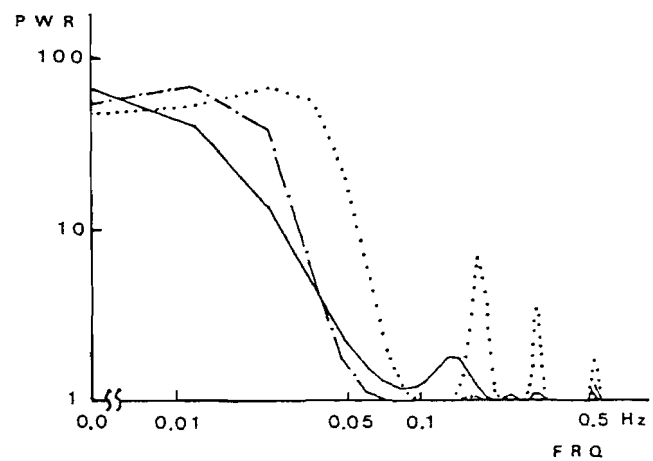


図 2-4 草丈のパワースペクトル

株径と草丈のデータが稲の生育状態を表すパラメータとして利用可能かどうかを調べるために散布図を描き、相関係数を求めた。これを図2-5に示す。相関係数 $r=0.19$ であり、ほぼ無相関であり、株径と草丈との関係は明らかにできなかった。なお草丈と重量とは相関のあることが報告⁵⁷⁾ されている。

次に操向系の入力となる稲列変位の変動範囲は-15.0~10.0cmであり、その変位量は25.0cmである。標準偏差がやや大きく少しばらつきが大きいですが、平均

値±12cm以内に約95%が含まれている。図2-6に稲列の変位を示す。稲列の曲がりは通常、田植機のオペレーション技術に依るところが大きい。これは人間による修正動作の結果でありその動作には周期性を伴うことが予想される。

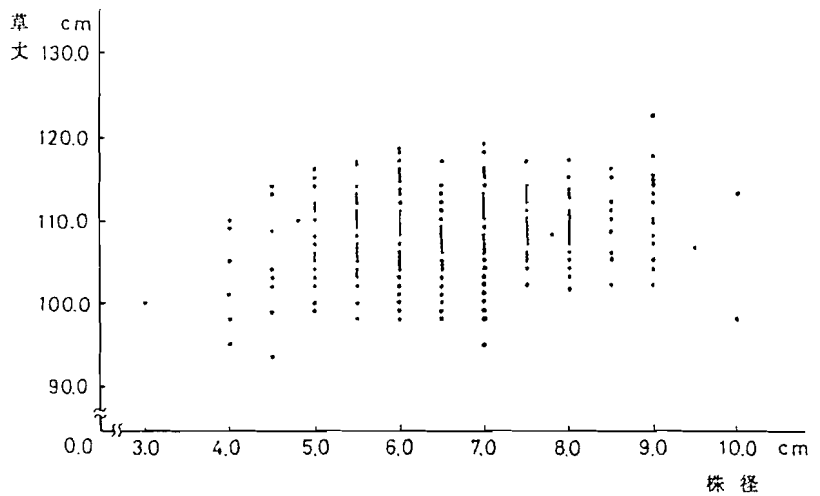


図 2-5 株径と草丈の散布図

そこで草丈と同様な条件で、スペクトル分析を行った。図2-7にオートコレログラム、図2-8にパワースペクトルを示す。これらの

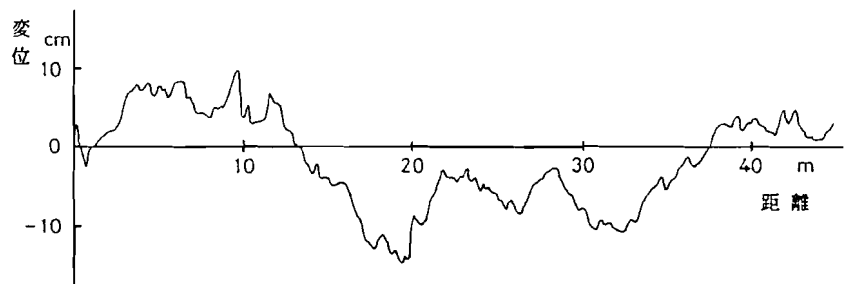


図 2-6 稲列変位

分析からは多くのランダム雑音の有する自己回帰性の性質を表す一次のマルコフ過程^{5,8)}と見ることができ、周期性を見出すことはできなかった。これは0Hzのパワーが最も強いことからデータ不足による長い周期成分の検出不能、極端な修正動作によるノイズ成分の増加あるいは田植機の走行経路が苗の成長が進行する間に变化して稲列の変位に反映されなかったことなどが原因として考えられる。

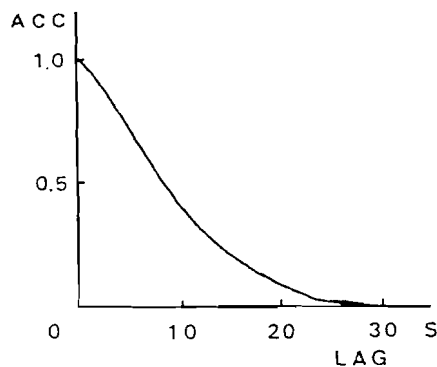


図 2-7 稲列のオートコレログラム

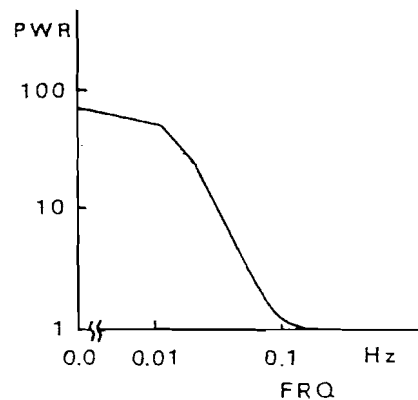


図 2-8 稲列のパワースペクトル

第 III 章 実験装置及び制御方法

自脱コンバインの制御は移動機械としての制御とコンバイン本来の機能の制御とに大別できるが、本研究では前者に重点をおくため走行各部を電氣的に制御可能とする必要がある。そこで制御対象として、左右の旋回を行う操向制御、前進、停止及び後進を行う変速制御、そして回行時に行う刈取り部の昇降制御の3系統とし、すべて油圧装置によって駆動するように供試コンバインを改造し、マイクロコンピュータによって制御した。本章では各制御装置の構成について述べ、また本研究の基本制御となる角速度センサによる旋回角検出について理論的及び実験的に考察した。

III-1 供試コンバイン

供試コンバインには、TC1410（ヤンマー農機（株）製造）を使用した。これは全面2条刈りの乗用自脱コンバインである。搭載エンジンはヤンマーディーゼルエンジン2TU-CAで、セルスタート方式水冷単気筒ディーゼルエンジンである。

変速装置として、パワートランスミッションを有し、クラッチ操作を必要とせず、レバー1本で変速が可能であり、副変速と合せて前進9段、後進3段の変速域がある。

走行装置はエンドレス・ゴム・クローラであり、その制御は操向クラッチと湿式ブレーキの併用である。

選別方式は、揺動、唐箕、吸引併用方式である。また、安全に作業を行うためのこぎ深さモータ警報装置、トップサッカ警報装置、2番スロワー詰り警報装置及びこぎ胴回転速度低下警報装置を具備している。

倒伏適応性は、向刈りの場合、倒伏角 70° 以下、追刈りの場合、倒伏角 85° 以下であり、左回刈りを基本とし、中割も可能である。

供試コンバインの全景を図3-1-(a)、図3-1-(b)に、仕様を表3-1に示す。



図 3-1-(a) 供試コンバインの全景 1



図 3-1-(b) 供試コンバインの全景 2

表 3-1 供試コンバインの仕様

形 式	YANMAR TC1410								
全 長 (mm)	2930								
全 幅 (mm)	1680								
全 高 (mm)	1650								
全 重 量 (kg)	950								
エンジン最大出力	14PS/2700rpm								
刈 幅 (mm)	770								
こぎ胴 直径×長さ (mm)	450×550								
走行装置	クローラ								
履 帯 幅 (mm)	330								
履 帯 長 (mm)	935								
操向装置	クラッチ アンド ブレーキ								
作業能率 (min/10a)	35-55								
各 変 速 位 置 別 速 度									
	前 進								
副 変 速	低			高			走 行		
パワーシフト	1	2	3	1	2	3	1	2	3
速 度 (m/s)	0.28	0.36	0.41	0.52	0.67	0.77	0.85	1.10	1.26
	後 進								
副 変 速	低			高			走 行		
パワーシフト	R			R			R		
速 度 (m/s)	0.28			0.52			0.85		

Ⅲ-2 各種制御装置

供試コンバインの制御装置は操向制御装置、刈取り部昇降制御装置及びパワーシフト制御装置の3系統から成立っており、それぞれ油圧装置を駆動源とし、ソレノイドバルブによって電氣的に制御可能とすることで、コンバイン各部の挙動をマイクロコンピュータによって自由に制御することができる。なおマイクロコンピュータ制御の有無に拘わらず、電氣的スイッチによって常に手動操作が優先する回路を各装置に組込んだ。本項では各種制御装置の構成及び役割について述べる。

Ⅲ-2-(a) 操向制御装置

コンバインが刈取り作業中に稲列に追従して進行するためにはオペレータは絶え間無い前方注視と頻繁なサイドクラッチ操作を強いられる。本装置はこの刈取り作業行程における経路修正の自動化（最短距離制御を含む）及びほ場末端部における回行の自動化を目的とした。

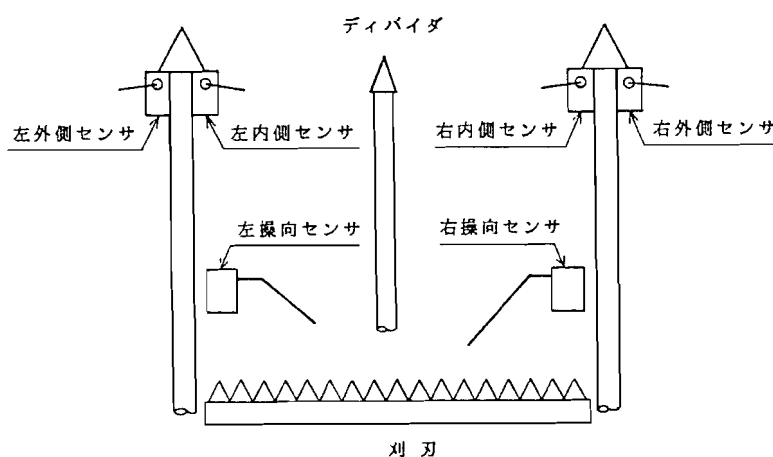


図 3-2 操向センサの配置

操向センサは左右のディバイダの先端に外側と内側それぞれ2個ずつ長さ7.5cmのローラアームを有する接触型リミットスイッチ（松下製 AZ8104）を取付けた。また左右のディバイダ内側に供試コンバイン専用の操向センサを2個取付けた。操向センサの配置を図3-2に示す。左側の供試コンバイン専用操向センサは縦刈り用センサであり、稲列との位置によって4種類の信号、「稲無し」、「右」、「直進」及び「左」を検出可能である。センサの構造は防水加工されたボックス内部にセンシングロッドにより回転するカムとその周囲に配置された3個のマイクロスイッチから構成されている。その構造を図3-3、また検

出信号の真理値表を表3-2に示す。

右側のセンサは横刈り用操向センサであり稲列との位置によって3種類の信号、「左」、「直進」及び「右」を検出可能である。その構造は左側のセンサと基本的に同様である。なお横刈り時は縦刈り時に比べて、株間が広いのでセンシングロッドを約5cmほど長くした。

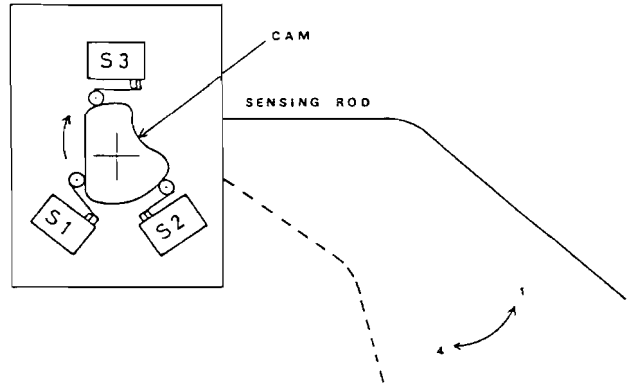


図 3-3 左操向センサの構造

表 3-2 左操向センサの真理値表

位置	S 1	S 2	S 3	備考
1	1	0	0	稲無し
2	1	0	1	右
3	0	0	1	直進
4	0	1	1	左

このセンサの構造を図3-4、また検出信号の真理値表を表3-3に示す。

供試コンバインは運動方向制御を操向レバーの操作によって操向クラッチを断続することにより行うため、アクチュエータとして専用のスプリング付単動の油圧シリンダを左右の操向クラッチに1個ずつ取付け、ソレノイドバルブによって制御した。

表 3-3 右操向センサの真理値表

位置	S 1	S 2	備考
1	0	1	左
2	0	0	直進
3	1	0	右

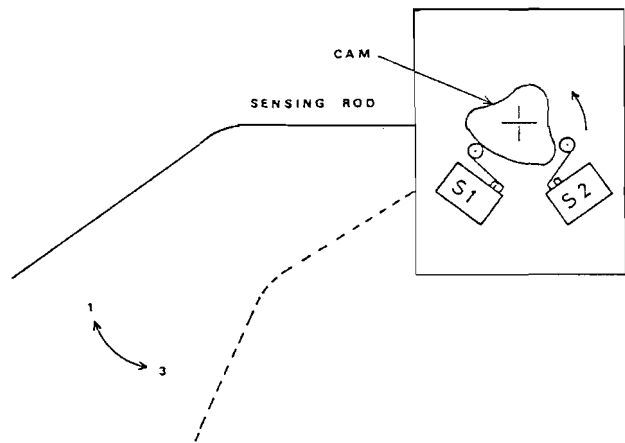


図 3-4 右操向センサの構造

Ⅲ-2-(b) 刈取り部昇降制御装置

本装置は回行時における旋回中に刈取り部の地面及び切株などへの接触防止を目的とするものであり刈取り作業行程における刈高さ制御は対象にしていな

い。

刈取り部の上下限検出に操向センサに用いたリミットスイッチを使用した。

刈取り部は油圧装置により駆動されるが電氣的に制御可能とするためコントロールバルブに換えてソレノイドバルブによって制御した。

Ⅲ-2-(c) パワーシフト制御装置

移動機械としてのコンバインを制御するために前進、停止、後進及び変速が自由に制御可能としなければならない。通常メカニカルなトランスミッションでは変速時にクラッチ操作が必要となり、この自動化は大変困難である。しかしながら現在普及しているトラクタ及びコンバインでは伝達効率はやや低いが、走行部を直接油圧駆動する無段変速が可能なHSTや有段ではあるが伝達効率のすぐれたパワーシフトトランスミッションを有した機械が多く、両者共クラッチ操作を必要とせずレバー1本で変速が可能であり自動化に都合のよい構造を有している。

供試コンバインにはパワーシフトトランスミッションが装備されており、その変速範囲は後進、中立、1速、2速、3速の5ポジションである。変速はパワーシフトレバーの直線移動によって行うため、これを操作するアクチュエータとして小型の複動油圧シリンダ(φ20×100mm)を使用した。油圧シリンダの暴走による変速位置検出センサの破壊を防ぐために、ストロークがパワーシフトレバーの移動範囲とほぼ等くなるように取付けた。また各変速位置での停止を確実にするためにチェック弁付流量制御弁を使用し、油圧シリンダの伸縮速度を調節した。

各変速位置の検出はフォトセンサを使用した非接触式の例⁴⁹⁾も見られるが、太陽光等の外乱による動作不良を避けるためには変調方式にするなど回路的に複雑化し、経済的にも不利となるため、

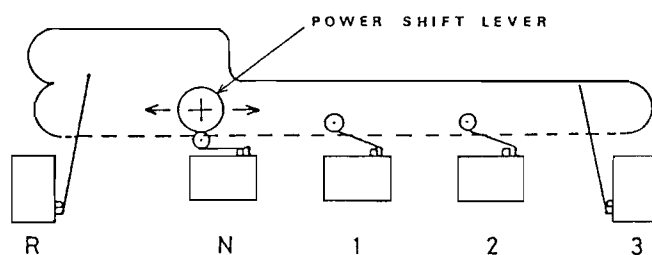


図 3-5 変速位置検出センサの配置

若干機械的強度は要求されるが小型で良質なマイクロスイッチによる接触式を採用した。この変速位置検出センサの配置を図3-5に示す。

Ⅲ - 2 - (d) 油圧装置

供試コンバインの各制御装置はすべて油圧装置によって駆動される。操向及びパワーシフト制御装置と刈取り部昇降制御装置の2系統から成立ち、それぞれ独立に制御可能となっているため操向しながら刈取り部を昇降するなど同時動作が可能である。

油圧装置は油圧タンク、ポンプ及びソレノイドバルブが一体となった供試コンバイン専用の装置を使用した。これは油圧シリンダへの配管以外すべて内部配管であるため非常にコンパクトとなっている。各装置には刈取り部昇降用のマニュアルバルブが取り付けられているがこれは使用しなかった。図3-6に油圧回路を示す。

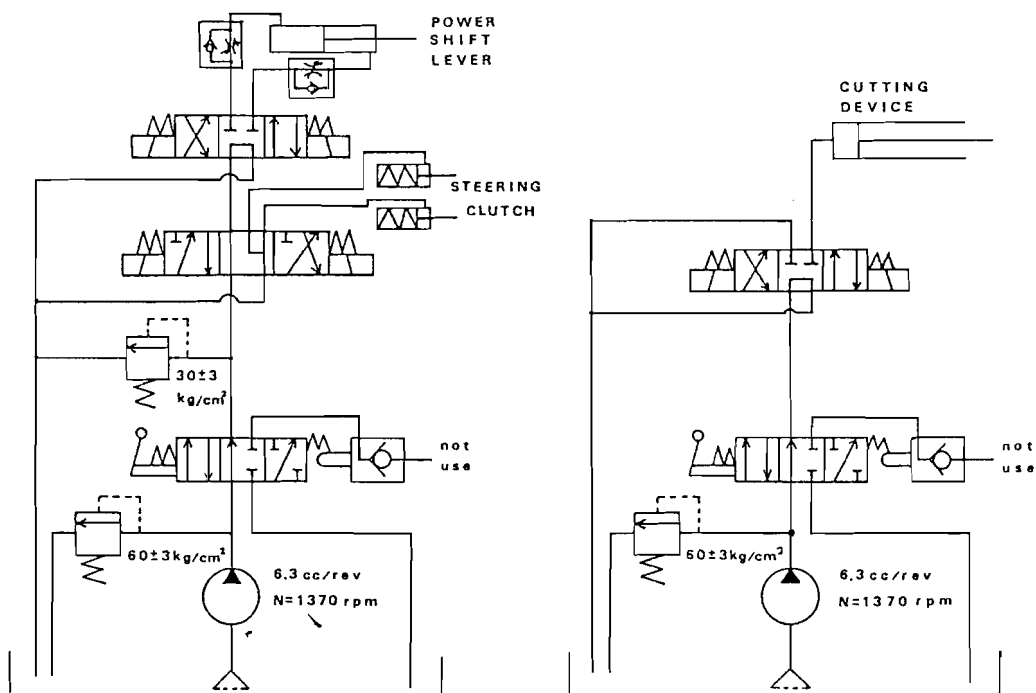


図 3-6 油圧回路

Ⅲ-3 電気制御装置

本装置は2つのマイクロコンピュータを中心に、それぞれ旋回角を演算するサブCPU部とコンバイン本体の制御を行うメインCPU部とから構成されており、メインCPU部はサブCPU部に対して旋回角データの転送要求を行い得られたデータにより回行制御及び最短距離制御を行う。本項ではこれら制御装置のハードウェアについて述べる。図3-7に本装置のブロック図を示す。

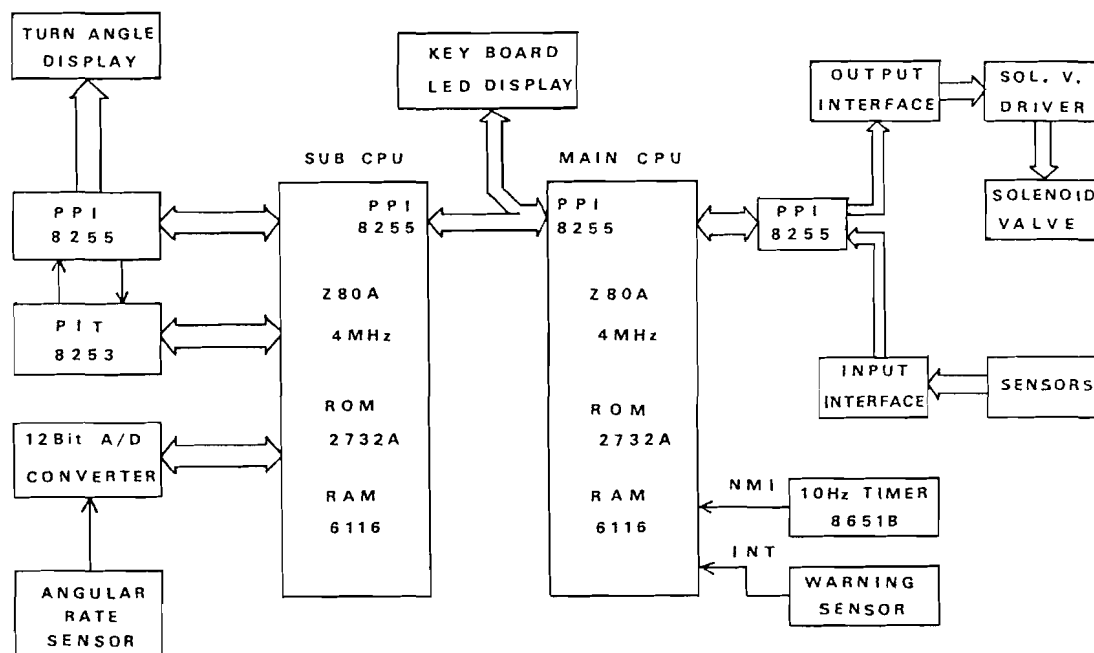


図 3-7 電気制御装置のブロック図

メインCPU部は各種センサからの入力情報の解析、各種油圧装置の駆動、サブCPU部からの旋回角情報の要求及び受信を行う。入力ポートとして、操向用センサ9bit、刈取り部昇降用上下限センサ2bit、変速位置検出センサ5bit、キー入力用8bit、また出力ポートとして、操向用、刈取り部昇降用及びパワーシフトレバー移動用の各ソレノイドバルブにそれぞれ2bit、その他サブCPU部とのハンドシェイクに11bit、リアルタイムクロック制御用に1bitを使用した。割込み入力として、0.1秒単位のタイマー割込みに「NMI」と初タンク満杯検出に「INT」をそれぞれ使用した。各ソレノイドバルブは自動走行中にも手動操作を可能とするために手動優先回路を設けた。

サブCPU部は角速度センサからのアナログ電圧をデジタル量に変換する12bit A/Dコンバータ(5mv/digit)、サンプリング周期信号を発生するP

I T (8253)、旋回角表示及びP I Tの制御を行うP P I (8255) とから構成されている。また旋回角情報の外部出力はP P Iのハンドシェイクによって行われる。

Ⅲ-3-(a) メインマイクロコンピュータ

本装置はC P Uに8bitのZ80A(クロック4MHz)を搭載し、R O M 4KByte、R A M 6KByte、I/Oポートとして24bitを備えた市販の制御用ワンボードマイクロコンピュータであるBC-800(大阪機電製)を使用した。本装置のブロック図を図3-8に示す。

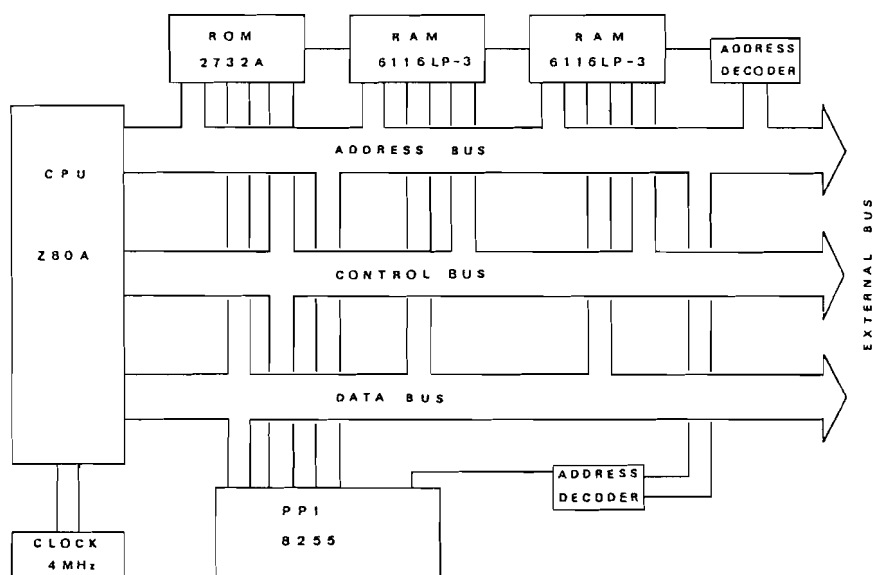


図 3-8 マイクロコンピュータのブロック図

実装メモリはデータ及びワークエリア用R A Mとして6116LP-3×3の6KByte、プログラムメモリとしてR O M(2732A)4KByteであるが、實際上、プログラム開発用として使用したR O MエミュレータがR O Mの代りとなった。

I/OポートはP P I (#0)が1個実装されているが、このポートはサブC P U部とのデータハンドシェイクポート、キー入力及び割込み制御用に使用した。

Ⅲ-3-(b) 入出力インターフェイス (メインCPU部)

マイクロコンピュータと外部機器を接続して外部との入出力を行う場合、TTLレベルに信号を合致させるインターフェイスを必要とする。本装置はPPI (#1) を1個増設し、入力ポートとして16bit、出力ポートとして6bit及びLED表示に2bitをそれぞれ割り当てた。入力ポートはすべて接点であるため、チャタリング防止回路を設けた。またノイズによる誤動作防止のためフォトカプラによる絶縁を行い、加えて接点側の電圧を12Vにすることによりノイズマージンの向上を計り、確実な動作を可能とした。出力ポートはすべてソレノイド駆動のためオープンコレクタタイプのバッファによりフォトカプラを駆動し、特に操向用ソレノイドバルブはパルスドライブとするため応答性及び信頼性向上を計りトランジスタドライブとした。この回路を図3-9に示す。

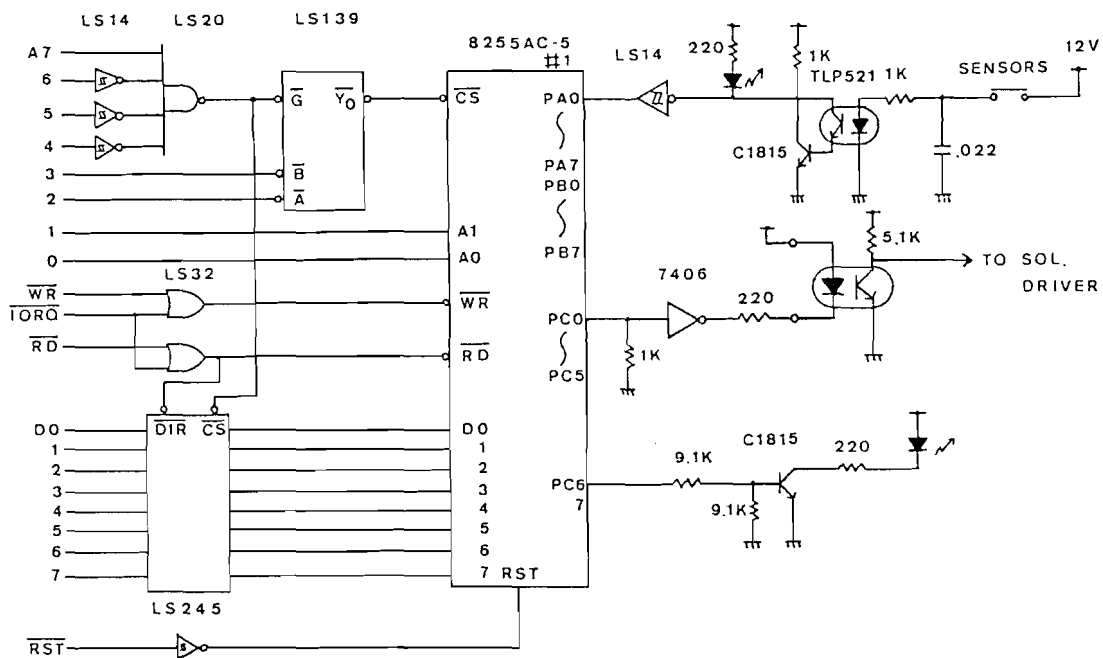


図 3-9 入出力インターフェイス回路 (メインCPU部)

油圧装置はソレノイドバルブによって制御されるため、電氣的スイッチによって手動操作可能とし、しかも緊急時の危険回避のため自動走行下でも常に手動操作を優先することが必要である。そこで出力ポートとソレノイドドライバの間に手動優先回路を付加した。この回路を図3-10に示す。

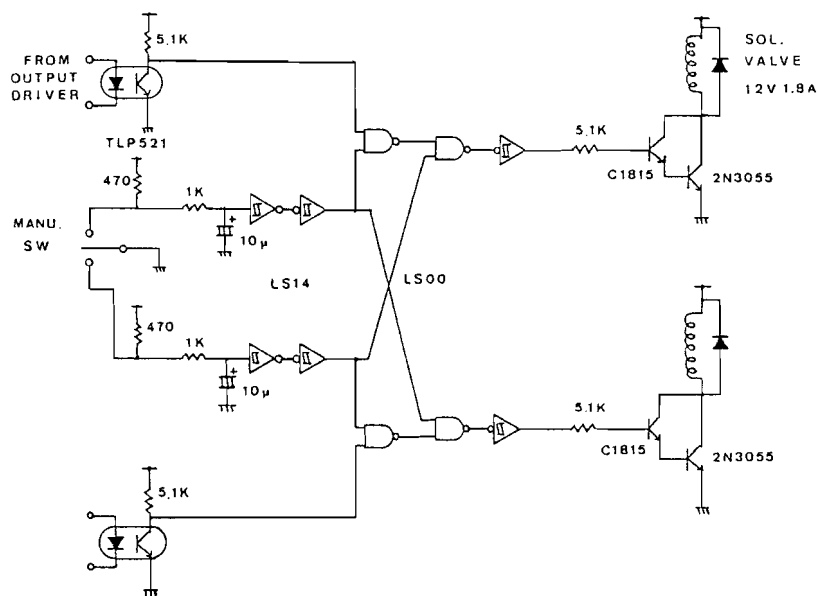


図 3-10 手動優先回路

本体の P P I (#0) は外部からの指令を受け付けるための 8 個のキースイッチ入力用、サブ C P U 部との通信用及び 10Hz タイマー I C の制御に使用した。通常キー入力はポーリングによって取り込むが、「B R K」キーは割込み信号「I N T」も発生させ、どんな状況下でもコンバインを停止させることを可能とした。またこの割込み信号は粉タンクの満杯を検出する信号にも使用し、同様にコンバインを停止する。この回路を図 3-11 に示す。

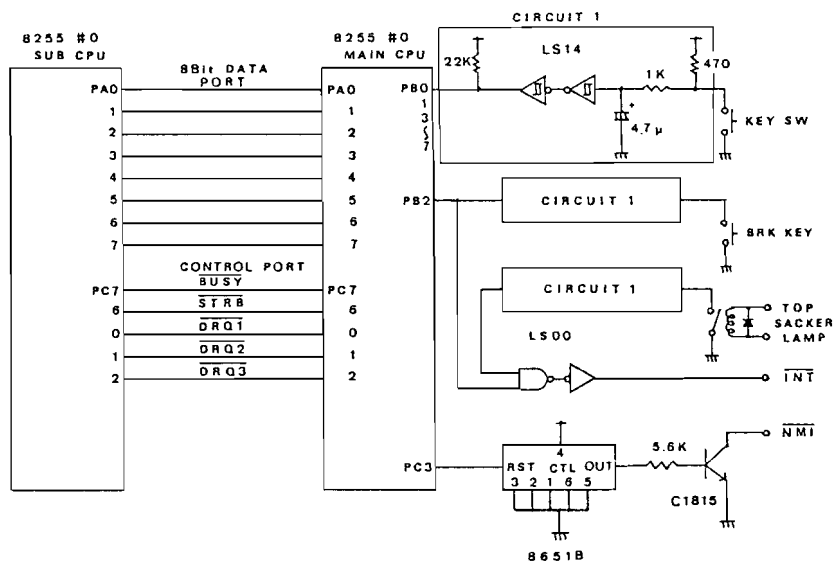


図 3-11 キー入力回路

メインCPU部のメモリマップ及びI/Oマップを表3-4に示す。

表 3-4 メインCPU部のメモリマップ及びI/Oマップ

メモリアドレス	機 能		
0000H-0FFFH	プログラムエリア	2732A	ROM
1000H-17FFH	データ及びワークエリア	6116LP-3	RAM
1800H-1FFFH	データ及びワークエリア	6116LP-3	RAM
3000H-37FFH	データ及びワークエリア	6116LP-3	RAM
I/Oアドレス	機 能		
00H-03H	ハンドシェイク及びキー入力用ポート		8255#0
80H-83H	センサ入力及び電磁バルブ駆動用ポート		8255#1

Ⅲ-3-(c) サブマイクロコンピュータ

サブCPU部の機能は角速度センサからの信号を処理して旋回角を演算及び表示することである。本装置にはメインCPU部に使用したのと同様の制御用ワンボードマイクロコンピュータを使用した。実装メモリはプログラム用ROM (2732A) に4KByte、ワークエリア用RAM (6116LP-3) の2KByteである。本装置には入出力インターフェイスカード及びA/D変換器カードが接続されるためバスラインにバッファを挿入し強化した。

Ⅲ-3-(d) 入出力インターフェイス (サブCPU部)

旋回角度の表示及びサンプリング周期を発生するPITの制御とその信号の入力にPPI (#1) を利用した。旋回角は符号付の4桁BCD表示とするため4bitを1桁とし、BCDデコーダにより7セグメントLEDを駆動した。PITはモード3で使用しデューティ比1の方形波を発生させ、その信号をPPIに入力し、ソフトウェアで波形のポジティブエッジを検出することでサンプリング周期を認識した。クロックはシステムクロックの2MHzを1/10分周し200

KHzを入力した。この回路を図3-12に示す。

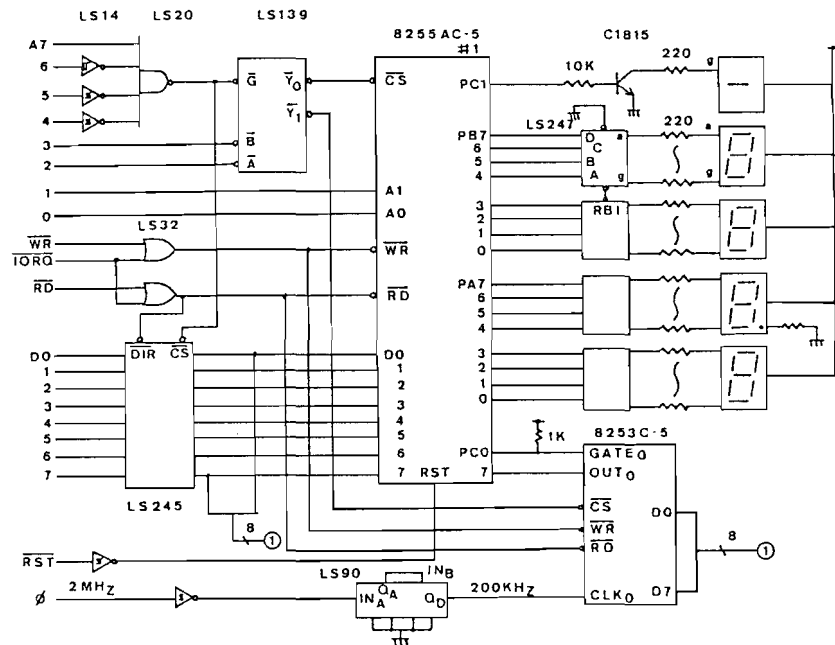


図 3-12 入出力インターフェイス回路 (サブCPU部)

III - 3 - (e) A/D変換器

角速度センサの出力は最大 $-12\sim 12V$ で、 $10mV$ の分解能を必要とするため $5mV$ の分解能を有する市販の12bitのA/D変換器 DAS-2812APC (サイエンス製) を使用した。A/DコンバータにパーブラウンのADC80-AG12、サンプル&ホールドにLF398を用いており、アキュジション・タイムの合計は $25\mu s$ である。入力は8chの差動入力でマルチプレクサにより切り換える。またノイズ防止のため $30Hz$ のローパスフィルタ (CR型) が組み込まれている。このA/D変換器のブロック図を図3-13に示す。また変換のタイミング・チャートを図3-14に示す。チャンネル番号を指定した後、マルチプレクサと後段のOPアンプの遅れ時間のため約 $20\mu s$ 待ち、スタートパルスを与える。そしてEOCがゼロになるまで約 $25\mu s$ 必要とし、その後上位バイトと下位バイトの2回に分けてデータを取り込む。このため最低 $45\mu s$ + 取込み時間を必要とし、チャンネルあたりの最大サンプリング周期は $70\mu s$ 程となる。

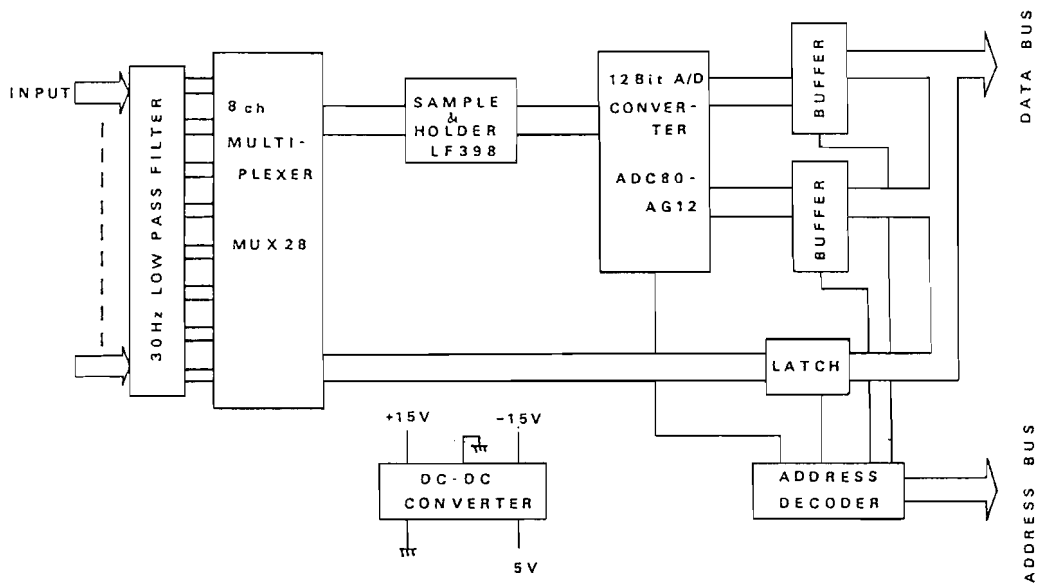


図 3-13 A/D変換器のブロック図

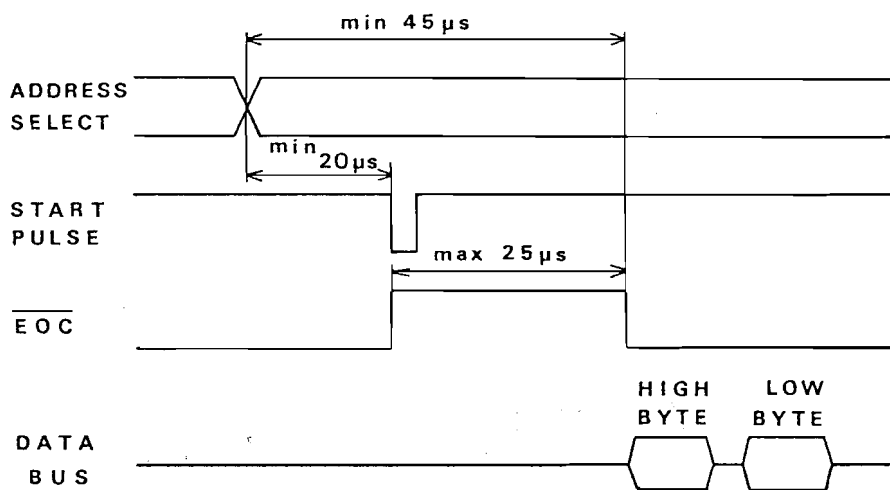


図 3-14 A/D変換のタイミング・チャート

サブCPU部のメモリマップ及びI/Oマップを表3-5に示す。

表 3-5 サブCPU部のメモリマップ及びI/Oマップ

メモリアドレス	機 能		
0000H-0FFFH	プログラムエリア	2732A	ROM
1000H-17FFH	データ及びワークエリア	6116LP-3	RAM
I/Oアドレス	機 能		
00H-03H	ハンドシェイク用ポート		8255#0
80H-83H	旋回角表示及びPIT制御用ポート		8255#1
84H-87H	サンプリング周期発生用タイマ		8253
90H-93H	12Bit A/D変換器		

Ⅲ - 3 - (f) 電源供給回路

マイクロコンピュータ及びその周辺機器は+5V、+12Vの安定した電源が必要である。コンバイン本体にはエンジン始動用その他に12Vのバッテリーが備えてあるため、これをソレノイドバルブの駆動に使用したが、マイクロコンピュータ用にはノイズ等による誤動作を防止するために12Vのバッテリーを増設して使用した。この12Vの電圧は定電圧ICにより+5V、また角速度センサの電源はDC-DCコンバータにより±15Vに変換して供給した。この回路を図3-15に示す。

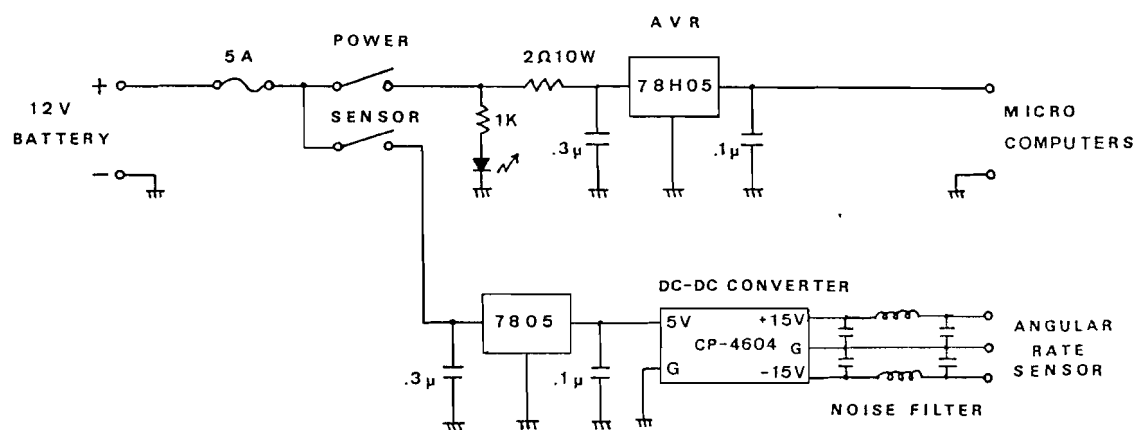


図 3-15 電源供給回路

Ⅲ－４ 旋回角検出方法

コンバインの旋回角、つまり地面に対する絶対的な旋回角をコンバイン自身が検出する方法として地磁気の利用、ジャイロスコープなどの慣性を利用する方法が考えられる。

本研究の初期に地磁気を利用した方向センサを考案し、その特性について検討した^{54, 55)}。このセンサは直接地磁気の3成分を検出するのではなく安価な船舶用磁気コンパスを利用し、その周囲の磁界の強度をマグネティックフラックスセンサで検出し間接的に地磁気の方角を知り、旋回角を検出するものである。しかしながら、振動及び傾斜の影響が大きく、小さい角度の検出には不適當であった。また地磁気を方向の検出に利用した自動車用のナビゲーションシステム⁵⁹⁾が実用化されているが、トンネル、鉄橋及び高圧線等の地磁気を乱す要因が多く精度はあまり高くない。

一方、ジャイロスコープはその回転軸が慣性空間に対して常に一定の方向を示す性質を持ち、加速度計と組み合わせて方向と移動距離とから位置を計算する慣性誘導装置 (INS) としてミサイル、ロケット及び飛行機等に応用されている。また2次元の方向の加速度を検知するガスレートジャイロを自動車の航法システム⁶⁰⁾に応用した例も報告されている。また最近では機械的なジャイロスコープに対してより精度の高い応答性の勝れたファイバジャイロ⁶¹⁾も実用化されている。このようにジャイロスコープは非常に勝れた方向センサであるが、その構造上高価なのが難点であり、コンバインへの応用には経済的に不利となる。

本研究では旋回角の検出に従来のレートジャイロと比較して安価であり、振動や衝撃にも強い特徴を持つ角速度センサを使用した。本項では角速度センサを利用した旋回角検出方法について述べる。

Ⅲ－４－(a) 角速度センサ

角速度センサは米国Watson社製でピエゾエレクトリック効果を持つ結晶体でできた片持ちばりの振動を利用し、測定軸まわりに生じる角速度に比例したア

ナログ電圧の信号と、回転方向の情報を共に発生するセンサである。これは従来のレートジャイロ스코ープに比較して、小型で正確、かつ振動や衝撃にも強く、その上小電力で作動しヒステリシスも無視でき、安価であるという特徴を持って

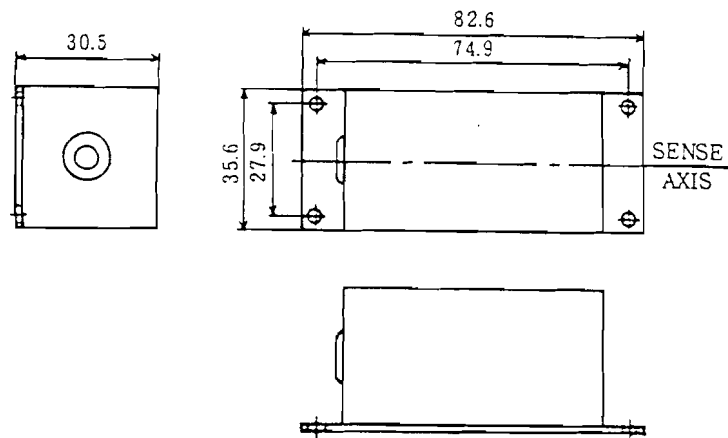


図 3-16 角速度センサの外観図

いる。しかしながら唯一の欠点として、若干ゼロ点の温度ドリフトを生じることである。この対策として使用前30分程度のウォームアップタイムを設けることとセンサ自体を断熱材で覆うことで対処した。角速度センサの外観図を図3-16、仕様を表3-6に示す。

表 3-6 角速度センサの仕様

入 力 電 源	±15V DC ±5%
出 力	角速度ゼロにて 0VDC 最大角速度にて ±10VDC
最 大 感 度	±100°/s
出 力 電 流	±10mA max
分 解 能	0.1°/s
精 度	2%
リニアリティー	0.1°/s
ヒステリシス	無し
加速度ドリフト	0.1°/s/G
周 波 数 応 答	70Hz
耐 衝 撃	200G 全方向
寿 命	10,000時間MTBF

III-4-(b) 角速度センサの性能試験

角速度センサの基本的な性能を確認するためには、正確な角速度を発生する装置が必要となり、角速度自身を比較検討することは非常に困難である。そこで原理的に一定の角速度が得られるステッピングモータを利用したターンテーブルを製作し、角速度センサを一定角度回転させ、その出力波形を一時デジタルメモリに記憶し、X-Yレコーダに記録した。そして、この出力波形の面積をディジタイザで測定して回転角を求め、設定値の回転角と比較検討した。

実験は回転角を 72° とし、角速度を $72^\circ/\text{s} \sim 108^\circ/\text{s}$ まで $9^\circ/\text{s}$ 毎に5段階に変化させたが、角速度の下限はステッピングモータコントローラの制限から $72^\circ/\text{s}$ としたためこの実験では $72^\circ/\text{s}$ 以下の角速度については実験不可能であった。

角速度センサの出力波形の一部を図3-17にまた実験結果を図3-18に示す。

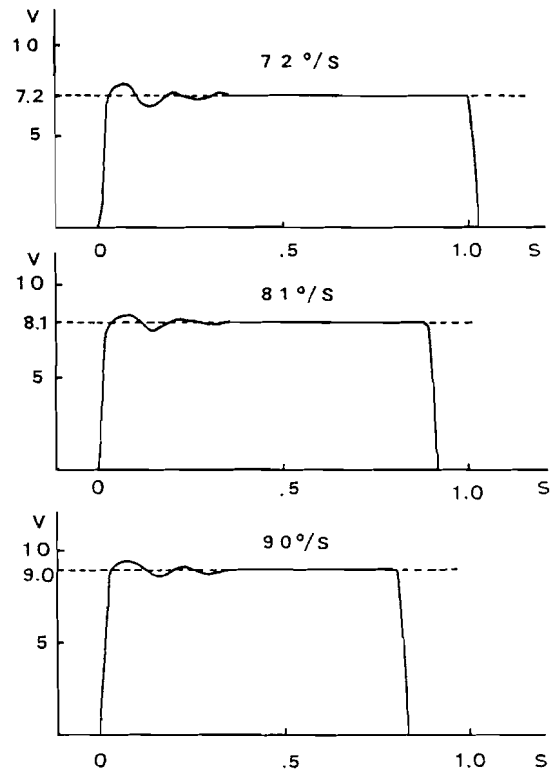


図 3-17 角速度センサの出力波形

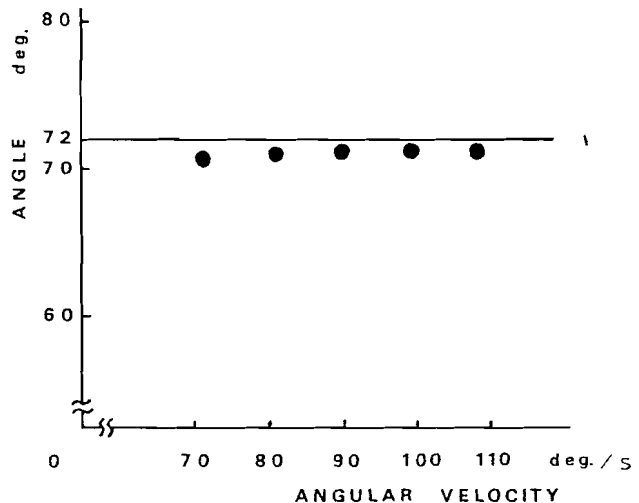


図 3-18 角速度センサの実験結果

Ⅲ-4-(c) 試験結果及び考察

ステップモータによる回転は1ステップ 1.8° であるため 72° の回転角を得るために40パルス必要となる。またパルスレートは $72^\circ/s \sim 108^\circ/s$ の角速度に対して40pps \sim 60ppsとなり、これは自起動周波数範囲内であるため加減速運転をせず一定パルスレートした。これによって角速度センサに与えられる入力にはステップ状となり、この実験は角速度センサのステップ応答を調べる意味を含んでいる。

角速度センサの出力波形を観察すると立ち上り部分にオーバーシュートが観られる。これを仮に角速度センサのステップ入力に対する過渡応答特性とすればその一巡伝達関数を2次振動系で近似し、減衰係数 ζ 及び固有角周波数 ω_n を行過ぎ量及び行過ぎ時間から実測し、求めると式(3-1)で表すことができる。但しゲイン定数は1とした。

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3-1)$$

$$\therefore \zeta = 0.62$$

$$\therefore \omega_n = 57.2 \text{ (rad/s)}$$

また周波数伝達関数 $G(j\omega)$ のボード線図を図3-19に示す。

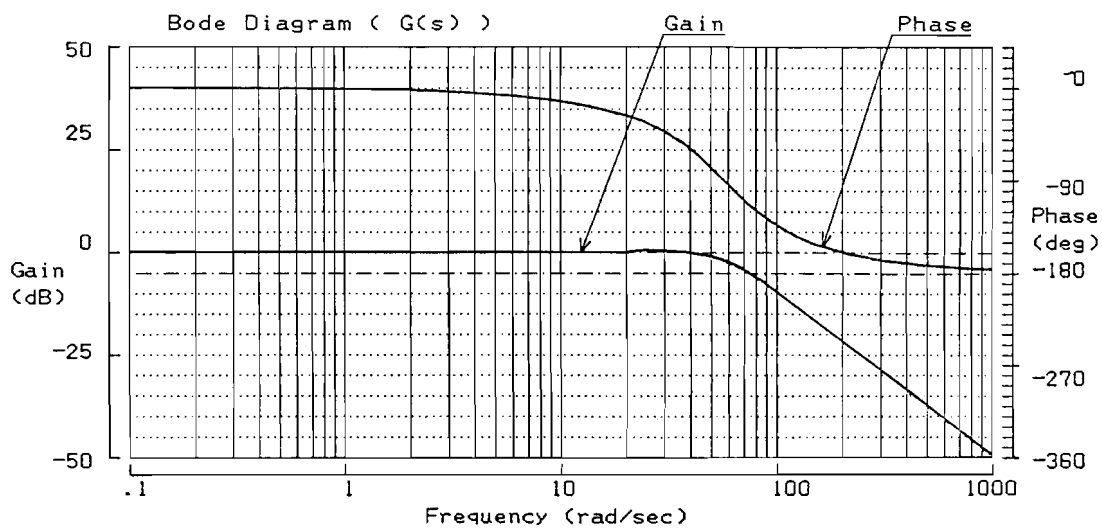


図 3-19 $G(j\omega)$ のボード線図

このボード線図からゲイン特性を調べると $\omega = 40$ (rad/s) 付近から下降しているのがわかる。これは角速度センサの周波数特性 (440 rad/s) から考えると非常に小さい値であるため、このオーバシュートは一概に角速度センサの過渡応答特性とは考えられず、起動時の角速度センサボックスの質量による慣性マスの影響及びステッピングモータの構造による低回転域での円滑でない回転などの影響の方が大きいと考えられる。これによってこの波形は角速度の変化そのものであると考えられ周波数応答は十分であることが確認できた。また実際の角度の理論値に対する誤差は最大で -1.8% 、平均約 -1.4% であり、カタログ上の仕様である 2% 以内に収まっていることが明らかになった。しかしながら、すべて理論値に対する誤差は負であり、やや低めの値を示す傾向がみられた。これは実験誤差、特にディジタイザによる面積測定に人間が介在するためによるものと考えられるが、このセンサに特有の特性と考えることもできる。

III - 4 - (d) 旋回角演算理論

旋回角は角速度を時間で積分することによって求められる。この理論積分値 $x(t)$ は角速度を $r(\tau)$ とすれば式 (3-2) で表される。

$$x(t) = \int_0^t r(\tau) d\tau \quad (3-2)$$

実際、マイクロコンピュータで積分するために数値積分を行わなければならない。数値積分の標準的な方法の一つは図3-20に示す2種の矩形近似である。その近似は幅 T の階段状波形の矩形面積を総合計することにより行われる。これらの矩形近似は図3-20に示したように各積分器の前へサンプルホールド装置を挿入することと等価である。図3-20(a)を矩形近似、(b)を進み矩形近似と言う。どちらも一長一短があるが安定度の点からここでは進み矩形近似を採用した⁶²⁾。なおもっと精密な数値積分は多角形ホールドの概念を使うことにより達成できるが、通常高次の積分法はシミュレーションモデルに不安定問題を生じる高次の伝達関数をもたらすため、制御システムではあまり使われない。

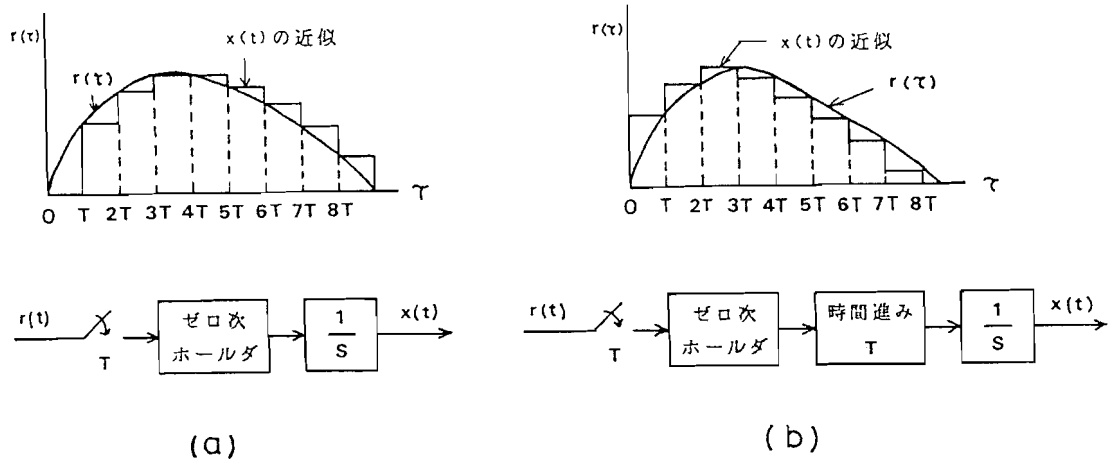


図 3-20 (a)矩形近似, (b)進み矩形近似

式(3-2)を離散的な形に置換えると数値積分値 I_n は式(3-3)で表すことができる。

$$I_n = \sum_{k=1}^n \omega(k) \cdot \Delta t \quad (3-3)$$

$$\because n = [t / \Delta t]$$

$\because \Delta t$: サンプルング周期

数値積分を行う場合、理論積分に対して誤差を伴うが、この誤差は入力周波数とサンプルング周期の関数と見ることができ入力周波数が低いほど、またサンプルング周期が短いほど少なくなる。入力波形を完全に再現するためにはナイキストのサンプルング定理により入力周波数の2倍以上の周波数でサンプルングする必要がある。この定理に従い角速度センサの性能を最大限に活用するためには、サンプルング周期を1/140ms以下にしなければならないため、8Bit CPUでは演算が困難である。しかしながら、入力周波数が低い場合にはサンプルング周期が大きくても積分誤差を小さくすることが可能である。そこで積分誤差とサンプルング周期、入力周波数の関係を調べるために周波数 f (Hz) のサイン波を入力波形とし、この角速度について $1/2 f$ の期間にわたって理論、

数値積分を行い、積分誤差について検討した。

入力角速度 $\omega = 10 \cdot V \cdot \sin 2\pi ft$ とすれば

ここで V は角速度センサの出力電圧

式(3-2)より理論積分値は式(3-4)となる。

$$I_t = 5 \cdot V (1 - \cos 2\pi ft) / \pi / f \quad (3-4)$$

また式(3-3)より数値積分値は式(3-5)となる。

$$I_n = 10 \cdot V \sum_{k=1}^n (\sin 2\pi f(k t_s)) t_s \quad (3-5)$$

$$\because n = [t / t_s]$$

$\because t_s$: サンプルング周期

よって積分誤差 E (%) は式(3-6)で表すことにする。

$$E = \frac{I_n - I_t}{I_t} \times 100 \quad (\%) \quad (3-6)$$

式(3-4), (3-5)よりサンプルング周期を50ms~500ms、入力周波数を0.2Hz~1Hzにわたって変化させたときの積分誤差を図3-21に示す。

このグラフから積分誤差は入力周波数が高いほど、またサンプルング周期が大きいほど急激に増加することがわかる。またどの入力周波数においても誤差がサンプルング周期によって増減していることがわかる。これはサンプルング周期と積分時間の関係によって

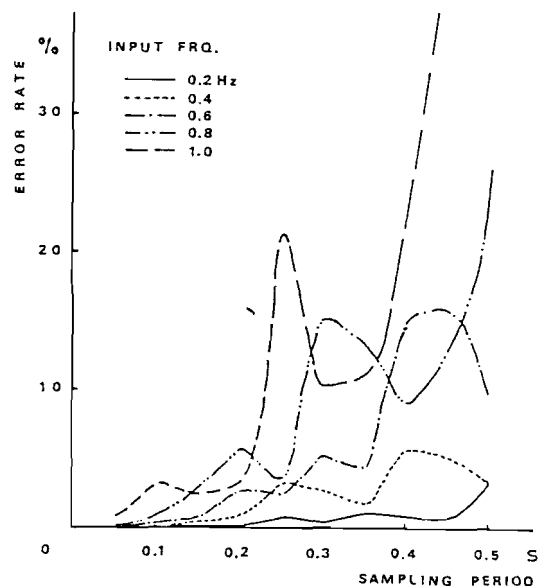


図 3-21 積分誤差

正負の誤差が打ち消しあうからであると考えられる。これらのことと演算時間の制約からサンプリング周期を200msとした。この場合、入力周波数1Hz以下で最大積分誤差6%以下に抑えられることが明らかになった。稲列の周波数分析の結果よりコンパインへの入力周波数は非常に低いため、サンプリング周期200msで実用可能であると言える。またエイリアシングを生じない最大入力周波数は理論上2.5Hzであるが、これ以上の周波数についてはゼロ次ホールドによるローパスフィルタ特性により軽減される。なお式(3-6)の計算結果はすべて負になったがグラフ上では正にして表現した。

Ⅲ-4-(e) 制御プログラム

式(3-3)による旋回角演算を中心としたサブCPU部のプログラムの概略フローチャートを図3-22に示す。プログラムはシステムの初期化後、サンプリング周期を発生するPITをスタートさせ、200ms経過後角速度センサの出力をA/D変換してCPUに取り込む。この値はコンピュータノイズにより2~3digitのバラツキを持つため、若干感度は下がるがゼロ点での誤差を防ぐために数値フィルタを設けて誤差の累積を防いでいる。このようにして変換された値はオフセットバイナリ形式のデータコードであるため、

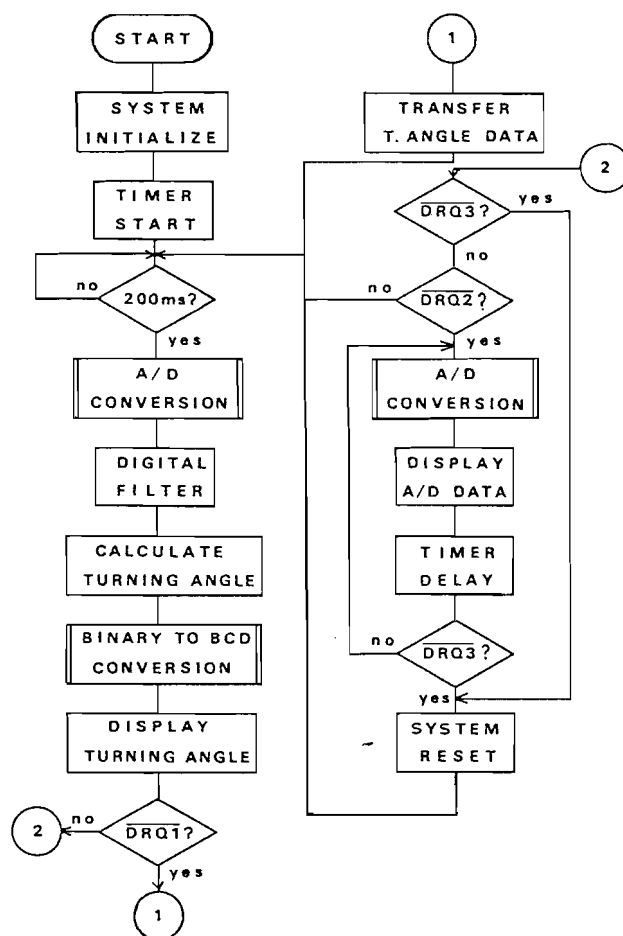


図 3-22 サブCPU部のプログラムの概略フローチャート

これを符号付バイナリ形式に変換して積分し200ms間の旋回角を求めメモリに積算する。またデータの取り出しを容易にするために8桁の符号付BCDコードに変換して、中位4桁を取り出し0.1°単位の値としてメモリにストアし、LEDディスプレイに表示する。最後に外部からのデータ要求を調べ、要求があれば旋回角データを出力する。(DRQ1) その他外部の要求として、角速度センサのオフセット電圧調整のためのA/D変換データの直接LED表示(DRQ2)及び計測開始のためのシステムリセット(DRQ3)がある。このようにして、200ms毎にこれらのルーチンを繰り返し、旋回角を演算する。なおプログラムの容量は約1Kbyte弱であり、全アッセンブルリストを付録に掲載した。

実際のサンプリング周期あたりの旋回角 A_t は式(3-7)で表される。

$$A_t = D_{ad} \cdot V_{ar} \cdot K_{av} \cdot T_{sp} \quad (^\circ) \quad (3-7)$$

D_{ad} : A/D変換データ (digit)

V_{ar} : A/D変換器の分解能 (V/digit)

K_{av} : 角速度センサの比例定数 ($^\circ/s/V$)

T_{sp} : サンプリング周期 (s)

ここで $D_{ad} = 1$, $V_{ar} = 5 \times 10^{-3}$, $K_{av} = 10$, $T_{sp} = 0.2$ とすれば

$\therefore A_t = 0.01^\circ$ となる。

これが演算上の最小単位となり、 $\pm 180^\circ$ まで表現すれば符号を含めて6桁の領域が必要となる。また $T_{sp} = 0.2$ としたため固定小数点法を用いれば乗算は全く必要とせず、実行時間の短縮が可能となった。

実際の旋回角演算過程を示すと

角速度センサの出力電圧の範囲は右旋回を正とすると

$$-10.24 \sim +10.235 \text{ V}$$

この電圧をA/D変換し2byteで表すと

$$0000 \text{ H} \sim 0FFF \text{ H}$$

これを符号付バイナリ形式に変換するため800Hを引くと

$$0F800 \text{ H} \sim 07FF \text{ H}$$

このデータを式(3-7)により角度に変換する。また積分されたデータは±180.00をしきい値として符号を逆転する。このしきい値をバイナリで表すと

0B9B0H ~ 4650H

このデータを表示するため6桁の符号付BCDコードに変換すると

818000 ~ 018000

ここで最上位の桁の「8」は負を表す。

実際に表示する桁数は0.1°単位の4桁であるため10倍して中位4桁を取り出し2byteの符号付BCDコードとしてメモリにストアする。

08180000 ~ 00180000
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
-1800 +1800

Ⅲ-4-(f) 旋回角演算装置の性能試験

旋回角演算プログラムの妥当性を検証するために旋回角演算装置の性能つまりマイクロコンピュータの積分精度を調べた。本装置の入力は電圧とその印加時間によって旋回角を演算するため、電圧は可変の定電圧電源、時間はPC8801により制御されるPITにより正確な時間を発生させて試験を行った。

電圧は0.5Vきざみ、印加時間は1秒及び2秒とし、電圧印加範囲は1秒間出力したのものは±10V間を、2秒間出力したものは旋回角が±180°の範囲でしか表示されないため、±8.5Vの範囲で測定した。なお試験は各電圧に対して10回繰り返しその平均をとった。

また3.6Vを10秒間出力すると理論上360°旋回(LED表示は0.0)することに相当するため20回の繰り返し試験を行った。

Ⅲ-4-(g) 試験結果及び考察

試験結果のグラフを図3-23に示す。回帰分析の結果、両者共ほぼ完全に直線上にのる結果が得られた。理論的にはサンプリング周期の倍数の印加時間であるため、内部的演算誤差は生じないが最小単位の0.1°の下桁を切り捨てるため±0.1°の誤差を生じる。

また3.6Vを10秒間出力した場合の20回の平均は0.0°であり誤差の範囲は±0.1°以内であった。

以上のようにほぼ理論通りの良好な結果と考えられる。

ここでサンプリング周期の倍数の印加時間でない場合の誤差について考えてみる。

入力電圧 V_{in} = 一定つまり角速度一定とした場合、ある時刻 t における旋回角 A_v は式(3-8)で表される。

$$A_v = K_{av} \cdot V_{in} \cdot t \quad (^\circ) \quad (3-8)$$

また旋回角演算装置による演算値 A_s は式(3-9)で表される。

$$A_s = K_{av} \cdot V_{in} \cdot [t / T_{sp}] \cdot T_{sp} \quad (^\circ) \quad (3-9)$$

これらの式をグラフ化した図を図3-24に示す。入力の角速度が一定であるため A_v は時間と共に直線的に増えていく。これに対して A_s は $t = n \cdot T_{sp}$ ($n = 0, 1, 2, \dots$) の場合には A_v と一致するが他の時刻では A_v に対して常に低い値となることがわかる。しかしながら、この誤差は一定のパターンで繰り返し、累積することはない。また V_{in} に比例して誤差の傾きも大きくなる。誤差の最大値は $E_a = K_{av} \cdot T_{sp} \cdot V_{in}$ となる。

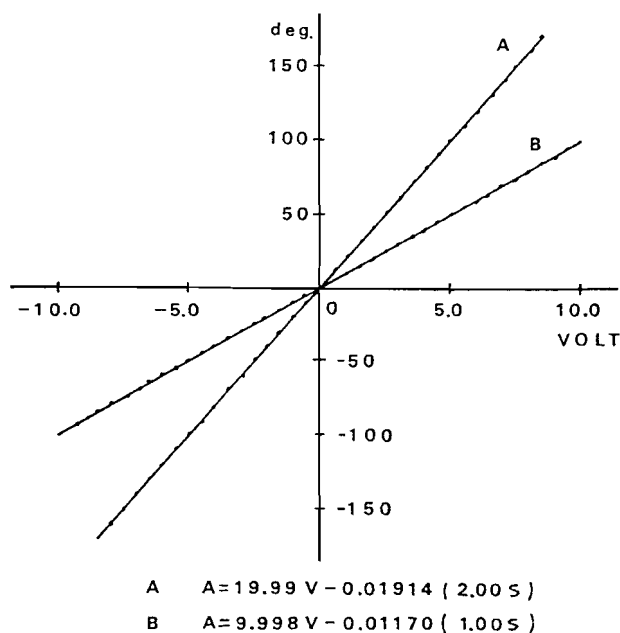


図 3-23 旋回角演算装置の積分精度

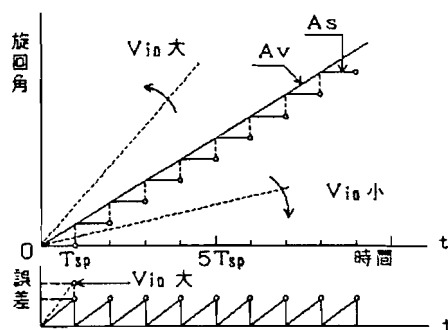


図 3-24 A_v と A_s の関係

今、 $V_{in}=10V$ の場合、 $E_a=20^\circ$ と大きな値となる。コンバインに生じる最大入力が行行時の旋回において生じるとすれば、進行速度 $V=0.5m/s$ 、旋回半径 $r=1m$ とすれば角速度 $\omega=V/r=28.6^\circ/s$ となる。よって $V_{in}=2.86V$ の場合 $E_a=5.72^\circ$ となる。Ⅲ-4-(d)において、入力周波数が低い場合、サンプリング周期が長くても誤差を小さくできると述べたが、 V_{in} の入力時間による誤差を伴い、これはサンプリング周期に比例するため、サンプリング周期はなるべく短いほうが、安定度の点で有効であると言える。仮にサンプリング周期を $0.02s$ とした場合、前述の誤差は 0.572° となり非常に高精度となる。現に航空機の姿勢制御にはこのサンプリング周期が用いられている⁶³⁾。

サンプリング周期の短縮のために現在のプログラムの実行時間について検討してみると、プログラムの実行時間は外部へのデータ要求処理を除いて、インターバルタイムカウンタによる実測で約 $28ms$ であった。データ要求処理は計算上数 ms であり、全体での実行時間は約 $30ms$ と予想される。しかしながら、実行時間の90%以上をBCD変換にかかっており、このルーチンは多分に冗長性を含んでいるため、アルゴリズムの変更により実行時間を短縮することが可能となるが、サンプリング周期を短縮することにより演算桁数も増加するため、より高速なアルゴリズムが必要となる。

Ⅲ-5 メインCPU部の制御プログラム

サブCPU部は角速度センサによる旋回角演算システムとして独立しており電源投入と同時に機能し、外部からの命令を監視しながら常に演算を続けている。これに対してメインCPU部はサブCPU部の管理とコンバイン本体の制御を行うシステムであり多くの機能を有している。

キーボードによって選択可能な機能は8種類ある。その機能を以下に述べる。

1. 通常自動走行・・・・・稲列に沿って自動的に刈取り作業を実行し、ほ場末端部では回行も行う。なお粉タンクが満杯になれば警報を発し、一時停止する。

2. 最短距離走行・・・収集した稲列データにより最短距離を演算し走行する。
3. 回行後進時間変更・・・回行時の後進時間をほ場状態により初期値を変更する。
4. 出力装置のテスト・・・操向装置、刈取り部昇降装置及びパワーシフト制御装置の動作試験を行う。
5. オフセット調整・・・角速度センサのオフセット電圧をゼロに調整する。
6. 稲列データ出力・・・稲列データを外部コンピュータに出力する。
7. 稲列データ収集・・・稲列に沿って自動的に刈取り作業を実行しつつ稲列データを収集する。
8. 自動回行・・・・・・・・ほ場末端部での回行を行う。

これらの機能は多くのサブルーチン等により実行される。以下にこれらのルーチンについて述べる。

1. キーインプットルーチン

制御装置のコンソールは5個の機能選択キーと選択した機能を実行する`START`キー、その機能を解除する`STOP`キー及び非常時にコンバインを停止する`BRK`キー等のキー群とキー選択の状態表示を兼ねた粉タンク満杯表示(TPF)と`BRK`キー押下表示(BRK)を行う2個のLEDとから構成されている。

電源投入時に2個のLEDを点灯しキー入力待ち状態になる。ここで機能選択キーを押すとBRKが消灯し、`START`キーにより実行されTPFも消灯する。また`START`キーを押す前に`STOP`キーを押すことにより再度機能の選択が可能となる。各機能の実行が終了すると2個のLEDを点灯しキー入力待ち状態になる。

2. 割込み処理ルーチン

本ルーチンはINT信号による割込み処理を行う。この信号はBRKキーと粉タンク満杯時に発生し、LEDの点灯(TPF, BRK)により両者の区別を行い、コンバインを一時停止する。STARTキーにより元の処理を再開する。

3. リアルタイムクロックカウンtrルーチン

本ルーチンは10HzのタイマーICによってCPUに対してNMIをかけることにより、常に0.1秒単位で時間カウントを行うことが可能である。メモリは2Byte使用し、BCD表現で0～999.9秒までカウントできる。なおこのルーチンの実行時間は最大33μsであり、他のルーチンへの影響はほとんどない。

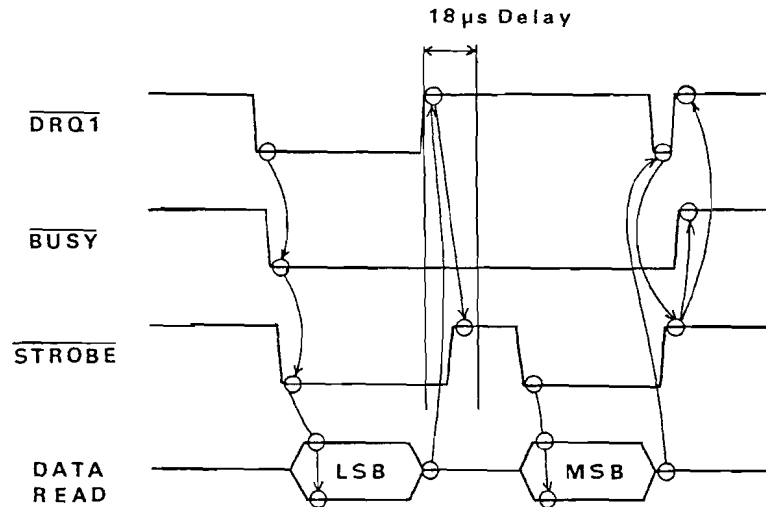
4. パワーシフト制御サブルーチン

本サブルーチンは現在の変速位置と指定された変速位置を比較し、その結果によってパワーシフトレバーを増速側か減速側に移動させ変速を行う。また変速位置を検出するセンサが油圧シリンダの過変位等により接触不良を生じた場合の対策として、変速位置のデータを記憶するメモリを1Byte設け、接触不良と判断した場合このメモリからデータを参照して変速を行う。このデータは変速を行う毎に更新される。従って、初期設定時に中立のデータをメモリに書き込むため、このルーチンを初めてコールする前にパワーシフトレバーを中立にしておく必要がある。

5. ハンドシェイクサブルーチン

本サブルーチンはサブCPU部からの旋回角データを確実に読み込むために下位バイト、上位バイトの順に1Byteづつタイミングを取りながらデータの授受を行うハンドシェイク法を採用した。図3-25に読み込みのタイミングチャートを示す。最初に旋回角データ転送要求信号のDRQ1をLにすると、サブCPU部は200msに1回のタイミングでこの信号を検出しBUSYをLにして信号の受付を示し、1Byte目のデータをセットしてSTROBEをLにする。そしてメインCPU部はこの状態を検出すると下位バイトを読み込み、DRQ1をHにして1Byte目のデータ転送を終了す

る。同様にして、
 DRQ1とSTR
 OBE信号により
 タイミングを取り
 ながら上位バイト
 の転送を終了する
 。なお $18\mu s$ のタ
 イムディレイは2
 Byte目のデータの
 受取を確実にする



ために設けてある。 図 3-25 データ読み込みタイミングチャート

6. サブCPU部リセットサブルーチン

サブCPU部は独立したシステムとして電源投入時から旋回角演算を続けている。従って、メインCPU部は任意の時点からの旋回角を必要する時、サブCPU部をゼロリセットする必要がある。本サブルーチンはサブCPU部に対してDRQ3を送出することによりゼロリセットを行うルーチンである。

7. A/D変換値表示要求ルーチン

角速度センサは若干温度ドリフトするためオフセット電圧をゼロ調整する必要がある。本ルーチンはサブCPU部に対して、DRQ2を送出することにより旋回角データを表示するLEDに角速度センサの出力電圧のA/D変換値を直接表示するルーチンである。

8. 刈取り部上昇サブルーチン

本サブルーチンは上限のリミットスイッチがONになるまでソレノイドバルブを励磁し刈取り部を上昇させるルーチンである。

9. 刈取り部下降サブルーチン

本サブルーチンは下限のリミットスイッチがONになるまでソレノイドバルブを励磁し刈取り部を下降させるルーチンである。

1 0 . 旋回角比較サブルーチン

本サブルーチンはハンドシェイクサブルーチンを利用して、設定旋回角を入力パラメータとして、現在の旋回角と比較するルーチンであり、設定旋回角を越えた時点でこのサブルーチンを抜ける。

1 1 . 出力装置テストルーチン

本ルーチンは出力装置である操向制御装置、刈取り部昇降制御装置及びパワーシフト制御装置の油圧装置やこれら装置に関連するセンサ等が正常に動作可能かどうかをチェックするルーチンである。但し、このルーチンには自己診断機能はなく、チェックは目視による。

1 2 . タイマーサブルーチン

動作の規定やセンサ入力時の遅延時間等にタイマールーチンが必要となる。本サブルーチンは0.05秒を基準に0.1、0.2、0.4、1.0、2.0秒の6種類のタイマーをソフトウェアによって生成しているため、それぞれサブルーチンとしてコールする。

1 3 . 稲列データ収集サブルーチン

本サブルーチンは最短距離制御における稲列の曲がりのデータを収集するルーチンであり詳細は第V章に譲る。

1 4 . 稲列データ出力サブルーチン

本サブルーチンは収集した稲列データを解析するためにPPIを通じて外部のコンピュータにデータを出力するルーチンである。

1 5 . 回行サブルーチン

本サブルーチンはほ場末端部で90°旋回するルーチンであり、詳細は次章に譲る。

1 6 . 右平行移動サブルーチン

本サブルーチンは自動走行中または回行後の稲列進入時に右外側センサが稲を検出した場合、右側に約35cm平行移動するルーチンである。センサが稲を検出するとその場で停止し、刈取り部を上昇しつつ、後進する。次に45°右旋回し1秒間後進後、45°左旋回し、停止する。そして刈取り部を

下降し前進する。このようにして右側に平行移動する。

17. 後進時間変更ルーチン

本ルーチンは回行時の後進時間をほ場状態によって初期値を変更するルーチンであり、キーボードによって0.5秒単位で増減可能である。

18. 最短距離走行ルーチン

本ルーチンは最短距離演算の結果に従い、最短距離走行するルーチンであり、詳細は第V章に譲る。

19. 最短距離演算サブルーチン

本サブルーチンは稲列データから最短距離ルートを演算し、目標点までの時間と方向をメモリに展開するルーチンであり詳細は第V章に譲る。

III-5-(a) 通常自動走行ルーチン

本ルーチンは稲列に沿って自動的に刈取り作業を実行し、ほ場末端部では回行を行い次の稲列に進入し、稲が無くなるまで自動走行するルーチンである。なお籾袋の自動交換は制御対象としていないため、籾タンクが満杯になれば一時停止する。また刈取りは縦刈りから実行する必要がある。これは縦刈りと横刈りでは稲の栽植密度が異なるため、稲の位置を入力する操向センサを変更しているためである。図3-26に本ルーチンの概略フローチャートを示す。

最初に約0.5m/sの速度で前進し、全操向センサを調べ右外側センサが稲を検出すると約0.4秒後に再度入力し、センサがONであれば初めて右平行移動ルーチンをセンサが稲を検出しなくなるまでコールする。このようにすべての操向センサの入力は一定時間後に再度入力することによってノイズ及び欠株による誤動作を防止した。

縦刈りか横刈りかを示すTRFLGフラグを参照して、それぞれの専用の操向センサを監視し、稲列に沿って左右旋回を繰り返しほ場端まで刈取り作業を実行して行く。図では横刈り時のフローチャートは入力する操向センサが異なる点を除いて同一であるため省略した。(B)また左右の操向用シリンダは機体のハンチングを防止するためパルス駆動し平均パワーを下げ制御した。

ほ場端の検出はすべての操向センサによる監視によって確実に検出することができる。検出後は左旋回に必要な距離前進し、回行サブルーチンをコールする。回行終了後、ストップフラグを調べ、刈取り終了かどうかを判断する。終了でなければTRFLGフラグを逆転して、以上のルーチンを繰り返す。

メインCPU部のプログラム容量は約3Kbyteであり、全プログラムのアッセンブルリストを付録に掲載した。

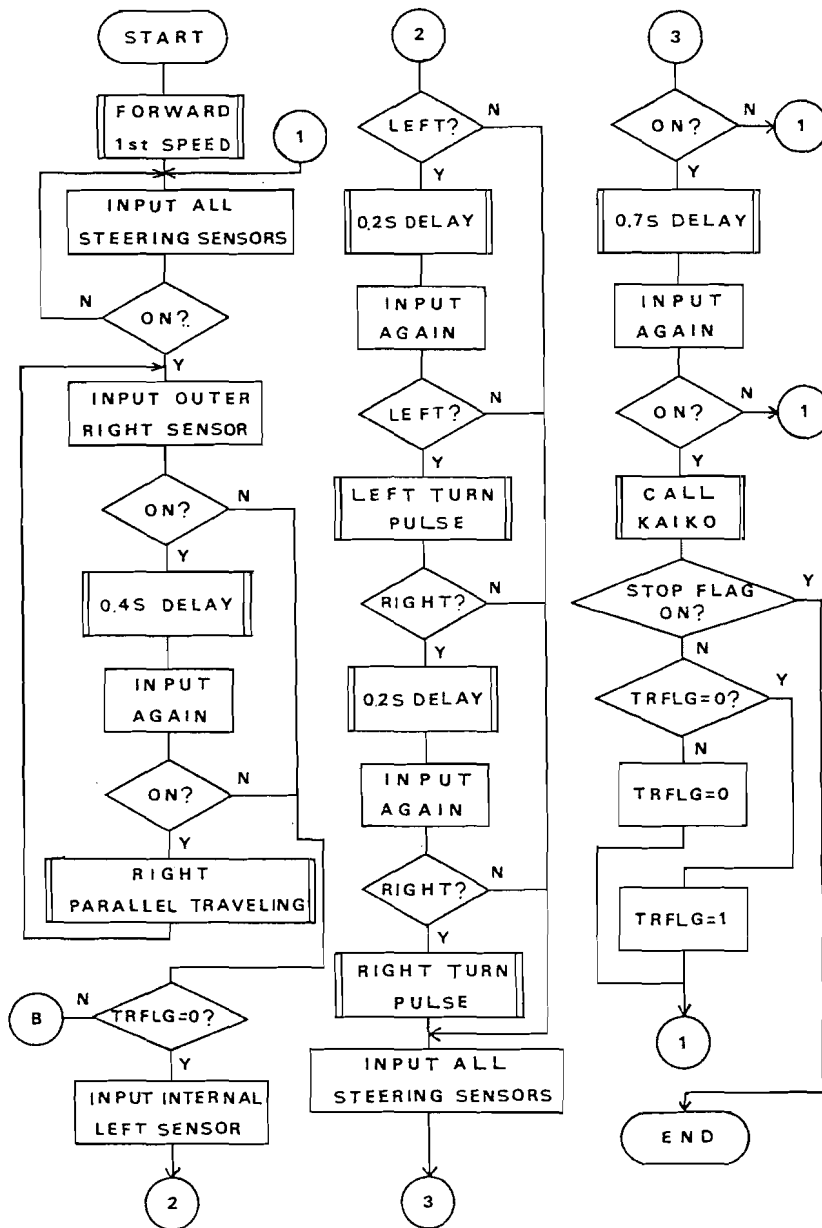


図 3-26 通常自動走行ルーチンの概略フローチャート

Ⅲ - 6 旋回実験

本研究の最も基本制御である角速度センサによって実際の旋回角を測定し、コンバインを任意の角度旋回させるフィードバック制御の精度及び振動による影響を検討するために実際のほ場において旋回実験を行った。実験は角度と方向をソフトウェアで設定し、ほ場においてコンバインを旋回させ、そのクローラの軌跡から実際に旋回した角度を測定した。

設定旋回角は 45° と 90° 、方向は右と左、速度は主変速と副変速の組み合わせで6段階に変化させた。また振動の影響を調べるために無負荷走行の場合と脱穀部を駆動して走行した場合とに分けて行った。旋回角の測定はクローラの軌跡から旋回前の直線と旋回後の直線を糸を付けた棒をこれらの直線に平行に挿入し、その交点の角度を分度器で測定した。また走行速度は5 m間の走行時間を各変速位置毎に測定して求めた。

Ⅲ - 6 - (a) 実験結果及び考察

設定旋回角が 45° 、 90° の実験結果をそれぞれ表3-7、表3-8に示す。

表 3-7 旋回実験結果 (45°)

設定角度	45°				平均 度
	右 旋 回		左 旋 回		
	走 行 の み	脱穀部駆動	走 行 の み	脱穀部駆動	
0.28	47	44	43	42	44
0.35	43	47	46	45	45.3
0.39	48.5	48.5	50	46	48.3
0.51	50	48	45	48	47.8
0.64	52	50	48	48	49.5
0.69	51	48	59	54	53
平 均	48.6	47.6	48.5	47.1	47.8

表 3-8 旋回実験結果 (90°)

設定角度	90°				平均 度
速 度 m/s	右 旋 回		左 旋 回		
	走 行 の み	脱 穀 部 駆 動	走 行 の み	脱 穀 部 駆 動	
0.28	90	97	87.5	95.5	92.5
0.35	90.5	94	85	85.5	88.8
0.39	83.5	84.5	94.5	89	87.9
0.51	94.5	97	96	91	94.6
0.64	99.5	100.5	91.5	93	96.1
0.69	97	102	95	94.5	97.1
平 均	92.5	95.8	91.6	91.4	92.8

以上の結果より45°の場合、その平均は47.8°であり旋回方向による差異はほとんどなく、また脱穀部駆動の有無による差異は脱穀部を駆動した場合が若干小さい程度でほとんどないと言える。また90°の場合、その平均は92.8°であり右旋回で脱穀部駆動時の場合がやや大きい、その傾向は45°の場合と似ていると言える。これらの傾向を確認するために繰り返しのある2元配置法による分散分析を行い、旋回角がどの因子に最も影響を受けるかを検討した結果、速度による影響が最も大きく、他

の因子はほとんど影響を与えないことが明らかになった⁶⁴⁾。そこで速度による影響を調べるために、設定角度別に回帰分析を行った。これらのグラフを図3-27、図3-28に示す。

45°の場合、危険率1%で回帰性が認められた。また単相関係

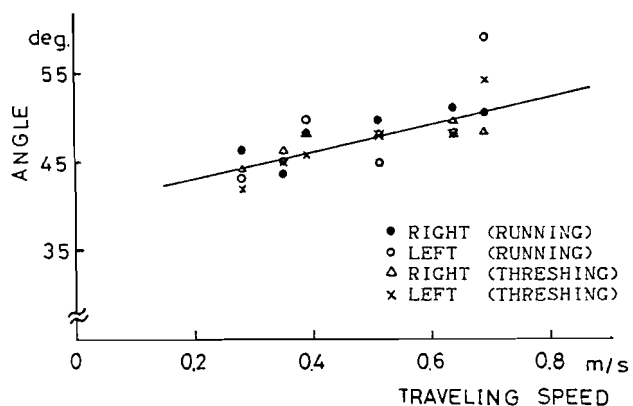


図 3-27 旋回実験結果 (45°)

数は0.72で無相関検定の結果、危険率1%で相関があるという結果が得られた。90°の場合についても危険率1%で回帰性が認められた。単相関係数は0.56で45°の場合と比べて小さい値であるが、無相関検定の結果、危険率1%で有意であるという結果が得られた。つまり速度の

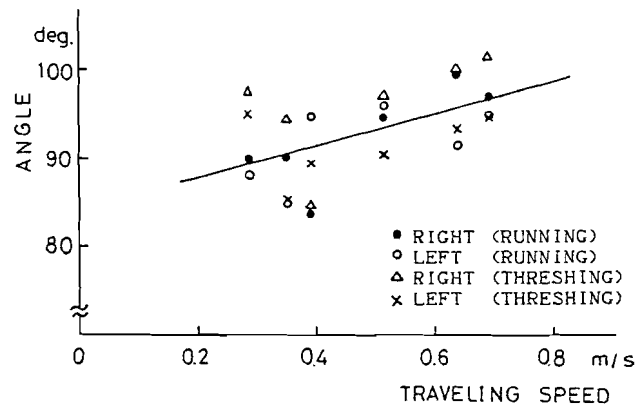


図 3-28 旋回実験結果 (90°)

増加に伴い旋回角が大きくなる傾向のあることが明らかになった。このような傾向を生じる主な原因は機体の慣性と油圧装置の伝達遅れである。この時間は速度に拘わらずほぼ一定であるため速度の増加に伴い、遅れ時間内に走行する距離が増加するためと考えられる。もう一つの原因はⅢ-4-(g)で述べたサンプリング周期と旋回時間の関係によって生じる誤差である。今、設定旋回角を $A_n(^{\circ})$ として各速度 $V_t(m/s)$ 別に設定旋回角に到達するまでの時間 $T_a(s)$ は旋回半径を $R_t(m)$ とした場合、式(3-10)で表される。

$$T_a = \pi \cdot R_t \cdot A_n / V_t / 180 \quad (s) \quad (3-10)$$

この T_a を式(3-9)の t に代入し、 $K_{av} \cdot V_{in}$ に角速度を代入すると A_n に到達した時点での旋回角演算装置の演算値が求まる。この数値は明らかに A_n 以下の値を取るため、実際の旋回角 $A_r(^{\circ})$ は A_n 以上の値になる。 A_r は式(3-9)を変形して式(3-11)で表される。

$$A_r = 180 \cdot V_t \cdot ([T_a / T_{sp}] + 1) \cdot T_{sp} / \pi / R_t \quad (^{\circ}) \quad (3-11)$$

なお R_t は旋回中心と左右クローラ間の中心との距離とし実測で1.1mであった。また実験結果の各速度毎の平均値と A_r をグラフ上に表し、速度による変

動の傾向を調べた。このグラフを図3-29、図3-30に示す。

このグラフから45°の場合、速度0.64m/sにおける角度がやや大きい点を除くと、ほぼ理論通りの傾向を示していることが明らかになった。また実際の旋回角は理論値に油圧系統の伝達遅れと機体の慣性の影響、それにゼロ点でのオフセットの影響を加えたものであるということが推察できる。

一方、90°の場合、実験値は低速度域においては理論値とまったく逆の傾向を示しているが、高速度域においては一定の間隔で非常に良く似た傾向を示している。特に90°の場合、45°に比

べて、旋回時間が長いこと角速度の変化の影響を受けやすいことなどが原因として考えられる。

以上のように設定旋回角に対する誤差は理論上生じる誤差に加えて、機体の走行速度に起因する誤差が最も大きいことが明らかになった。これは旋回時間がサンプリング周期の倍数になる旋回角度を除いて、設定旋回角度旋回することが理論上不可能であることを示している。本研究ではパワーシフト1速、副変速Hの速度を設定しているが、これは旋回実験における0.51m/sに相当する。この速度におけるTaを式(3-10)から計算すると $T_a = 0.038 \cdot A_n$ となる。このTaが0.2の倍数となる旋回角は式(3-12)で表される。

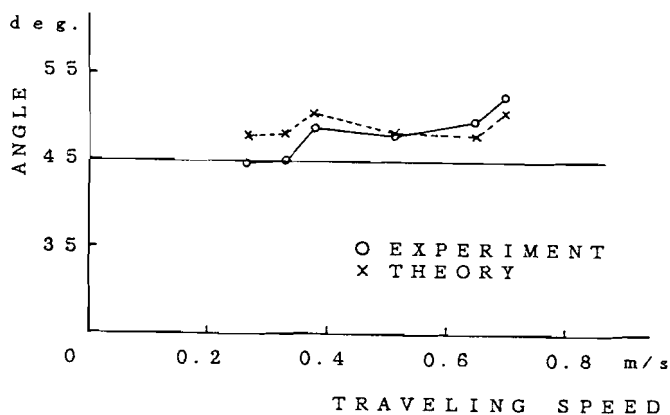


図 3-29 実験値と理論値との比較(45°)

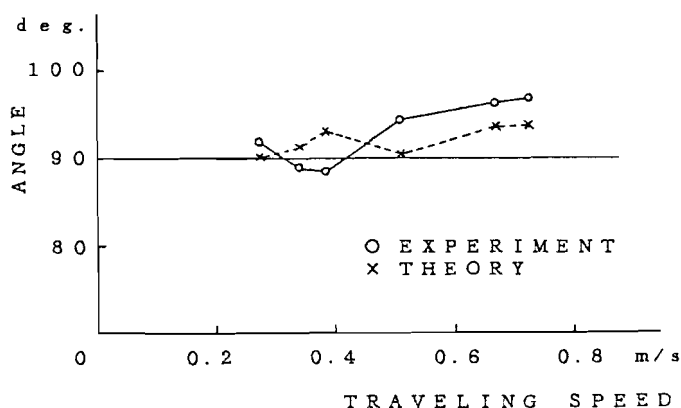


図 3-30 実験値と理論値との比較(90°)

$$A_n = 0.2 \cdot n / 0.038$$

$$A_n \doteq 5.26 \cdot n \quad (^\circ) \quad (n = 0, 1, 2, 3, \dots) \quad (3-12)$$

この設定角の時に理論上の誤差はゼロになるが、実際には前述のような誤差が伴うため、この速度について実際に旋回させ、角度設定を適切な値にすることで誤差を最小限に押えることが可能となると考えられる。

III-7 旋回角検出装置の理論的解析

旋回角検出装置はコンバインの旋回角速度を検出し、デジタルコンピュータにより積分して旋回角を計算し、設定旋回角との差をゼロにするようにコンバインを旋回させる一種のデジタル制御系と見ることができる。

本制御系のブロック線図を図3-31に示す。

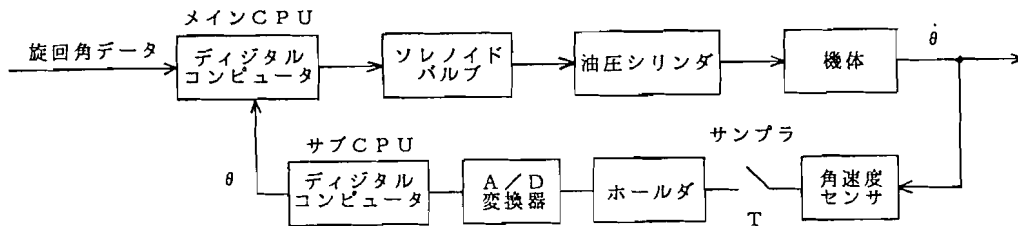


図 3-31 旋回角検出装置のブロック線図

本制御系は基本的に論理演算を含む非線形制御系であり正確な伝達関数を求めることは不可能である。そこで各要素を線形近似して伝達関数を求めた。このブロック線図を図3-32に示す。各要素は線形近似されているが、これはサンプリングを持つサンプル値制御系であり線形解析を行うにはz変換法⁶⁵⁾を適用する必要がある。

以下に各要素の伝達関数を求めた。

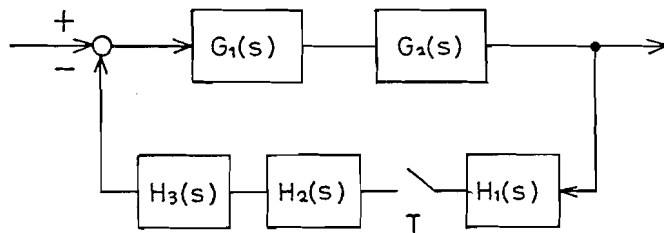


図 3-32 線形近似したブロック線図

ソレノイドバルブ及び油圧シリンダは比例要素（ゲイン K_1 ）及び一次遅れ要素（時定数 T_1 ）で近似した。この伝達関数を $G_1(s)$ とすると式(3-13)で表される。

$$G_1(s) = \frac{K_1}{1 + T_1 s} \quad (3-13)$$

コンバイン本体は剛体であると仮定すれば、簡単な質量または慣性モデルとして表せる。機体の慣性モーメントを J とすれば加えられるトルクと速度との間の伝達関数 $G_2(s)$ は式(3-14)で表される⁶⁶⁾。

$$G_2(s) = \frac{1}{J s} \quad (3-14)$$

角速度センサは比例要素（ゲイン K_2 ）で近似した。この伝達関数を $H_1(s)$ とすれば式(3-15)で表される。

$$H_1(s) = K_2 \quad (3-15)$$

ゼロ次ホールダの伝達関数 $H_2(s)$ はサンプリング周期を T とした場合、式(3-16)で表される。

$$H_2(s) = \frac{1 - e^{-Ts}}{s} \quad (3-16)$$

A/D変換器とサブCPUは単純に比例要素（ゲイン K_3 ）と積分要素で近似すれば伝達関数 $H_3(s)$ は式(3-17)で表される。

$$H_3(s) = \frac{K_3}{s} \quad (3-17)$$

前向き伝達関数 $G(s)$ は式(3-18)で表される。

$$G(s) = G_1(s) G_2(s) = \frac{K_1}{J s (1 + T_1 s)} \quad (3-18)$$

フィードバック伝達関数は式(3-19)で表される。

$$H(s) = H_1(s) H_2(s) H_3(s) = \frac{K_2 (1 - e^{-Ts})}{s^2} \quad (3-19)$$

システムの閉ループ z 変換伝達関数 $W(z)$ は式(3-20)で表される。

$$\begin{aligned} W(z) &= \frac{G(z)}{1 + G(z) H(z)} \\ &= \frac{\mathfrak{Z} [G(s)]}{1 + \mathfrak{Z} [G(s) H(s)]} \end{aligned} \quad (3-20)$$

ここで記号 \mathfrak{Z} は z 変換を示す。

式(3-20)各要素は式(3-21)及び式(3-22)となり、これらを式(3-20)に代入すると式(3-23)となる。

$$\begin{aligned} \mathfrak{Z} [G(s)] &= \mathfrak{Z} \left[\frac{K_1}{J s (1 + T_1 s)} \right] \\ &= \frac{K_1}{J} \mathfrak{Z} \left[\frac{1}{s (1 + T_1 s)} \right] \\ &= \frac{K_1}{J} \frac{z(1 - e^{-T/T_1})}{(z-1)(z - e^{-T/T_1})} \end{aligned} \quad (3-21)$$

$$\begin{aligned} \mathfrak{Z} [G(s) H(s)] &= \mathfrak{Z} \left[\frac{K_1}{J s (1 + T_1 s)} \frac{K_2 (1 - e^{-Ts})}{s^2} \right] \\ &= \frac{K_1 K_2}{J} \frac{z-1}{z} \mathfrak{Z} \left[\frac{1}{s^3 (1 + T_1 s)} \right] \end{aligned}$$

$$\begin{aligned}
&= \frac{K_1 K_2}{J} \frac{z-1}{z} \mathfrak{Z} \left[\frac{1}{s^3} \right] - T_1 \mathfrak{Z} \left[\frac{1}{s^2} \right] + T_1^2 \mathfrak{Z} \left[\frac{1}{s(1+T_1 s)} \right] \\
&= \frac{K_1 K_2}{2J} \frac{(z-e^{-T/T_1}) \{T^2(z+1)-2T_1 T(z-1)\} + 2T_1^2(z-1)^2(1-e^{-T/T_1})}{(z-1)^2(z-e^{-T/T_1})} \quad (3-22)
\end{aligned}$$

∴ W(z)は

$$\begin{aligned}
&= \frac{2K_1 z(z-1)(1-e^{-T/T_1})}{2J(z-1)^2(z-e^{-T/T_1}) + K_1 K_2 \left\{ (z-e^{-T/T_1}) \{T^2(z+1)-2T_1 T(z-1)\} + 2T_1^2(z-1)^2 \right\}} \\
&\quad \frac{1}{(1-e^{-T/T_1})} \quad (3-23)
\end{aligned}$$

このように制御系の伝達関数を z 変換手法により求めたが、実際の制御系には非線形要素や論理演算を含むため、この伝達関数によって制御系の特性を解析することには若干無理がある。本来この制御系はマイクロコンピュータをコントローラとしているために、システムの解析にはコンピュータによるデジタルシミュレーション手法が非常に有効となる。

Ⅲ-8 旋回実験のシミュレーション

システムの挙動を解析的に行うことが不可能であるため、デジタルコンピュータによるシミュレーションが有効となる。またシステムパラメータを容易に変更可能なため、実験不可能な状態を再現することも可能である。本項では旋回角検出装置の解析を含めた旋回実験のシミュレーションを行った。サンプリング周期の変更によるシステムの挙動の変化を検討した。またコンバインの振動による影響を調べるために振動成分の周波数分析を行い、シミュレーションに加えた。

Ⅲ-8-(a) 供試コンバインの振動解析

角速度センサは本来外部振動の影響をキャンセルする構造を有しているが、周波数、振幅及び位相の異なる多くの振動成分を同時にキャンセルすることは困難である。特にコンバインは振動の発生源となる往復運動機構が多いこと及び機体の剛性が余り高くないことなど振動の多い機械と言える。

そこで供試コンバインの振動の周波数成分を調べるために角速度センサ取付け部に加速度計をヨー軸方向に固定し、ストレインアンプによって増幅後、データレコーダに記録した。そしてこのデータをスペクトラムアナライザによって周波数分析した。実験はコンバインを停止状態とし、エンジン回転速度を作業時に設定し、①エンジン回転のみ（E）、②脱穀部駆動（E + T）

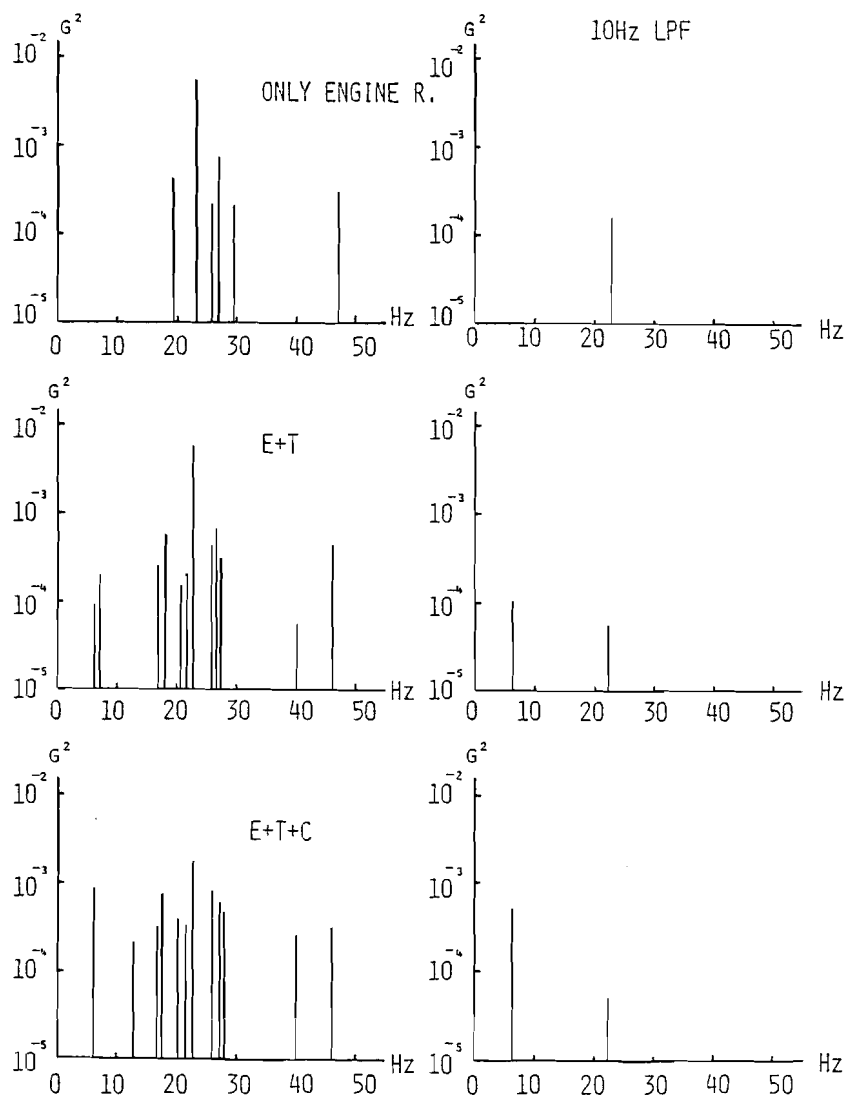


図 3-33 供試コンバインの振動分析結果

、③脱穀部+刈取り部駆動（E + T + C）の条件毎に行った。この結果を図3-33に示す。またシミュレーションの一つのパラメータとするために、角速度センサの出力電圧の周波数分析を行い、実際に受ける影響を明らかにした。なおエンジンの回転速度は約2700rpmであった。

エンジン回転速度から換算して46Hzのスペクトルがエンジンの振動によるものであることがわかる。またどの条件下でも、その約半分の周波数である23Hz付近のスペクトルが最も大きいことがわかる。これはエンジン出力が一旦ベルトによって半分に減速されたカウンタプーリ軸の振動であると言える。つまりこの軸に各部の駆動源が集中しているため最も大きな振動源になったと考えら

れる。また各部を駆動した場合カッター、脱穀部及び揺動選別部等の往復運動によって振動成分の増加することがわかる。右のグラフはストレインアンプ内蔵の10Hzのローパスフィルタを挿入した場合のスペクトルを示しているがカウンタプーリ軸のパワーが1/10程度減衰し、他は低周波付近を除いてほぼゼロになっている。

次に角速度センサの出力の周波数分析の結果を図3-34に示す。前述のように振動分析の結果から23Hz付近のスペクトルのパワーが最も大きいことが明らかになったが、この図から角速度センサが最も影響を受ける振動源はカウンタプーリ軸ではなくエンジンであることが判明した。これは角速度センサの構造上の特性によって、特定の周波数成分が減衰するためであると考えられる。

以上の結果から振動の周波数分布が明らかになったが、これは停止状態での分析であるため、走行時には路面状態による振動の影響を考慮する必要がある。これは約4Hzの周波数であることが報告されている⁶⁷⁾。

このように角速度センサは振動による影響を受けるため、何等かの対策をする必要がある。最も容易な手段として、以上の周波数分析から約4Hz以上の周波数成分をカットするローパスフィルタの挿入が考えられる。しかしながら、かなり急峻なカットオフ特性を必要とし、必然的に高次のフィルタとなるため立ち上り特性と整定時間による時間遅れが原波形に歪を生じさせてしまう原因となる。これは実際カットオフ周波数2Hzの4次のベッセル型ローパスフィル

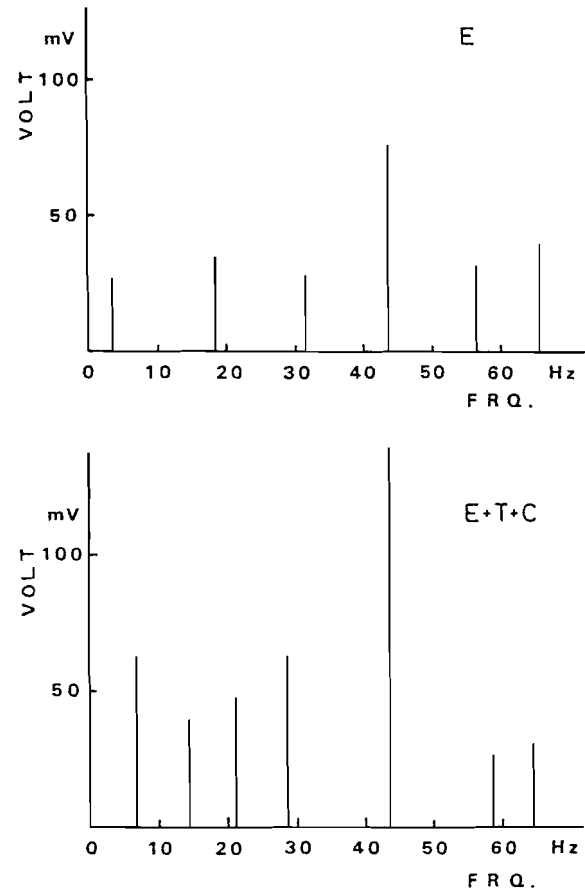


図 3-34 角速度センサの出力の周波数分析

タを挿入した場合、設定旋回角に対して実際の旋回角がかなり大きくなることを確認した。しかしながら、旋回実験から、旋回時においてはそれほど振動の影響を受けないことが明らかであるため、ローパスフィルタの挿入はむしろ逆効果であると言える。本研究における角速度センサの応用は旋回時に限られ、直進時における計測は行わないため、ローパスフィルタ無しで十分応用可能であると考える。

III-8-(b) コンバインの走行モデル

コンバインの走行装置は履帯方式であり、その操向は片側の履帯をブレーキングするいわゆるスキッドステアリング方式である。このような装軌式車両の旋回理論は比較的古くから研究⁶⁸⁻⁷⁵⁾されているが、本来すべりを伴う操向方式であるため走行部の形状及び路面の性状に大きく左右されるため、理論的確立が困難であると言える。そこでこのシミュレーションにおけるコンバインの走行モデルは単純な幾何学的方法によって表現し、機体の慣性、走行部のすべり及び路面の性状はすべて無視

した。図3-35に走行のモデルを示す。コンバインは中央ディバイダの先端P (p , q)、左右の旋回中心 $O_r (a , b)$ 、 $O_l (c , d)$ 及びこれらの交点M (e , f) で代表した。これによってコンバインの進行方向を直線PMの勾配で表現することができる。なお旋回中心は左右の履帯の接地部を2分する線上にあると仮定し、また旋回半径は実測して求めた。

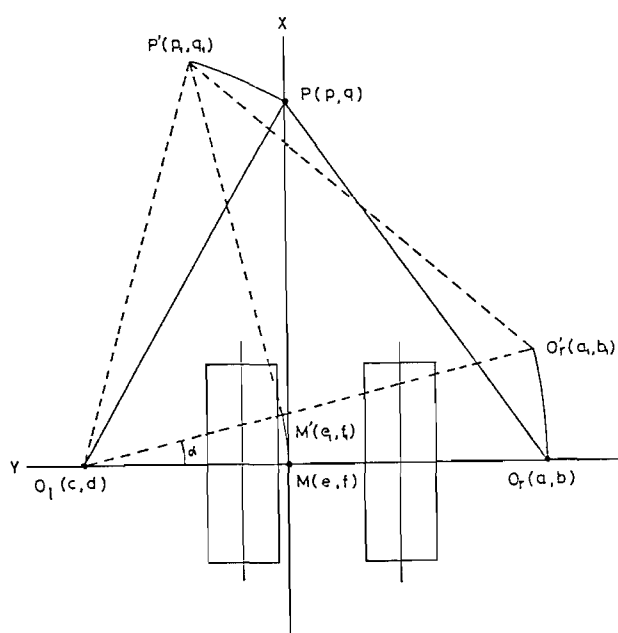


図 3-35 コンバイン走行のモデル

III - 8 - (c) 機体移動の概念

コンバインの移動は幾何学的座標変換によってシミュレートするが、ここでコンバインの走行速度を v (m/s) として t 秒間移動した場合の走行距離は $v t$ (m) となり、左旋回角 α (rad) は旋回半径を L とすると $\alpha = v t / L$ (rad) と表せる。今、機体が α (rad) だけ左旋回した場合の P 、 O_r 、 M の旋回後の位置をそれぞれ P' 、 O_r' 、 M' とすれば各点の座標は式(3-24)で表される。

$$\begin{aligned}
 p_1 &= (p-c)\cos\alpha - (q-d)\sin\alpha + c & \alpha \geq 0 \\
 q_1 &= (p-c)\sin\alpha + (q-d)\cos\alpha + d \\
 a_1 &= (a-c)\cos\alpha - (b-d)\sin\alpha + c & (3-24) \\
 b_1 &= (a-c)\sin\alpha + (b-d)\cos\alpha + d \\
 e_1 &= (e-c)\cos\alpha - (f-d)\sin\alpha + c \\
 f_1 &= (e-c)\sin\alpha + (f-d)\cos\alpha + d
 \end{aligned}$$

また機体が β (rad) だけ右旋回した場合は式(3-24)と同様に各点の座標は式(3-25)で表される。

$$\begin{aligned}
 p_1 &= (p-a)\cos\beta - (q-b)\sin\beta + a & \beta \leq 0 \\
 q_1 &= (p-a)\sin\beta + (q-b)\cos\beta + b \\
 c_1 &= (c-a)\cos\beta - (d-b)\sin\beta + a & (3-25) \\
 d_1 &= (c-a)\sin\beta + (d-b)\cos\beta + b \\
 e_1 &= (e-a)\cos\beta - (f-b)\sin\beta + a \\
 f_1 &= (e-a)\sin\beta + (f-b)\cos\beta + b
 \end{aligned}$$

機体の直進運動は各座標点が進行方向に対して平行に移動することによって表現することができる。コンバインの進行方向は点Pと点Mを結ぶ直線で表すことができ、この直線の勾配が機体の向きを決定する。今、速度 v (m/s) で t 秒間移動した場合の各点の x 方向成分と y 方向成分のそれぞれの増分 x_s 、 y_s は式(3-26)で表される。

$$\begin{aligned} x_s &= (p-e) \cdot vt / \sqrt{(p-e)^2 + (q-f)^2} \\ y_s &= (q-f) \cdot vt / \sqrt{(p-e)^2 + (q-f)^2} \end{aligned} \quad (3-26)$$

従って、各座標は式(3-27)で表される。

$$\begin{aligned} p_1 &= p + x_s \\ q_1 &= q + y_s \\ \\ a_1 &= a + x_s \\ b_1 &= b + y_s \\ \\ c_1 &= c + x_s \\ d_1 &= d + y_s \\ \\ e_1 &= e + x_s \\ f_1 &= f + y_s \end{aligned} \quad (3-27)$$

制御システムの特性を表すために装置に関する諸数値を基礎実験により求めた。これを以下に示す。

油圧装置の伝達遅れ時間

右操向系 OFF→ON : 0.07 s、 ON→OFF : 0.14 s

左操向系 OFF→ON : 0.07 s 、 ON→OFF : 0.12 s

旋回半径

右 : 1.05 m 、 左 : 1.1 m

コンバインの各点の初期座標は P (0,0) とした場合

O_r (-170,-120) 、 O_l (-170,95) 、 M (-170,0)

III - 8 - (d) 旋回シミュレーション実験

旋回実験と同様なパラメータでシミュレーション実験を行った。時間間隔は 0.01秒とし、各時刻におけるコンバインの位置を計算し、最終停止位置における座標から旋回角を求めた。また各パラメータの内「走行のみ」と「脱穀部駆動」の区別は振動解析における結果を利用した。これは各条件における振動成分がサイン波として角速度センサの出力に重畳されたものとして扱った。これを表3-9に示す。シミュレーションは BASIC

表 3-9 振動の周波数と振幅

項 目	振幅 (mV)	周波数 (Hz)
走 行 の み	75	43.5
	39	65.4
	35	19
脱 穀 部 駆 動	133	43
	64	21.5
	63	6.3
	45	28.8
	39	14.6

言語で記述し実行はPC-9801Fで行った。プログラムの概略フローチャートを図3-36に示す。またプログラムリストを付録に掲載した。

プログラムは旋回角、「走行のみ」か「脱穀部駆動」かを入力して、まず各パラメータから0.01秒間に移動する距離と旋回角を計算して初期設定を終える。

次に時間カウンタTSをゼロにして、2秒間コンバインを直進させる。そしてサブCPU部をリセットするが、このリセットには最高、サンプリング周期分かかるため、この時間までの乱数を発生させてリセット時間を決定した。従って、この時間と油圧装置の伝達遅れ時間だけコンバインは余分に直進するこ

とになる。

このようにして旋回動作に移行し、0.01秒間にこの旋回によって発生する角速度を計算し、この角速度を角速度センサの出力電圧に変換する。そしてコンバインの位置を計算し、画面にプロットする。またTSがサンプリング周期の倍数の時刻の時、A/D変換して旋回角を計算し、設定旋回角を越えたら、旋回を中止して直進移動を2秒間行い停止する。この移行時には油圧装置の伝達遅れ時間だけコンバインは余分に旋回する。

そして最終停止位置の座標から旋回角を計算して、各設定条件と共にプリンタに印刷する。これを走行速度0.28~0.69m/sまでの6段階を順番に実行して、一つの実験区を終了する。なお振動成分の初期時刻は乱数で決定した。

またサンプリング周期を0.05秒とした場合についても同様な実験を行い、0.2秒の場合と比較検討した。

Ⅲ-8-(e) 実験結果及び考察

このシミュレーション実験の結果を表3-10~表3-13に示す。

実際の実験と比較して、数値はやや大きいバラツキが小さい他は速度に対する相関は強いと言える。左右旋回による差異、「走行のみ」、「脱穀部駆動」による差異もほとんどなく、実際の実験同様の傾向を示していると言える。振

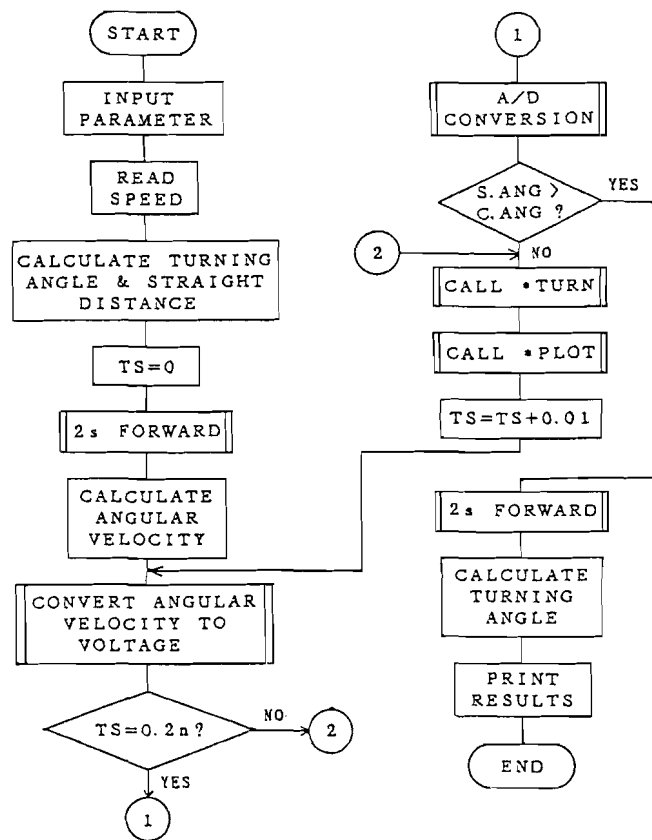


図 3-36 シミュレーションプログラムの概略フローチャート

動による影響が少ないのは振動によって生じる電圧が旋回によって生じる電圧に比べて十分小さいためであると考えられる。

表 3-10 旋回シミュレーション実験結果 1

設定角度	45°				サンプリング周期：0.2秒
速 度 m/s	右 旋 回		左 旋 回		平 均 度
	走行のみ	脱穀部駆動	走行のみ	脱穀部駆動	
0.28	47.5	47.5	48.2	48.2	47.9
0.35	47.9	47.9	49.4	49.4	48.7
0.39	49.1	49.1	50.9	50.9	50.0
0.51	53.1	53.1	50.7	50.7	51.9
0.64	52.6	52.6	50.2	50.2	51.4
0.69	49.2	56.7	54.2	54.2	53.6
平 均	49.9	51.2	50.6	50.6	50.6

表 3-11 旋回シミュレーション実験結果 2

設定角度	90°				サンプリング周期：0.2秒
速 度 m/s	右 旋 回		左 旋 回		平 均 度
	走行のみ	脱穀部駆動	走行のみ	脱穀部駆動	
0.28	93.3	93.3	92.0	92.0	92.7
0.35	93.7	93.7	93.1	93.1	93.4
0.39	95.9	95.9	95.6	95.6	95.8
0.51	97.5	97.5	93.1	93.1	95.3
0.64	94.5	94.5	96.8	96.8	95.7
0.69	94.3	101.9	97.2	97.2	97.7
平 均	94.5	96.1	94.6	94.6	95.1

表 3-12 旋回シミュレーション実験結果 3

設定角度	45°				サンプリング周期：0.05秒
速 度 m/s	右 旋 回		左 旋 回		平 均 度
	走行のみ	脱穀部駆動	走行のみ	脱穀部駆動	
0.28	47.5	47.5	46.8	46.8	47.2
0.35	47.9	47.9	46.6	46.6	47.3
0.39	48.0	48.0	47.9	47.9	48.0
0.51	48.9	48.9	48.0	48.0	48.5
0.64	49.1	50.9	48.6	48.6	49.3
0.69	49.2	51.1	48.8	50.6	49.9
平 均	48.4	49.1	47.8	48.1	48.3

表 3-13 旋回シミュレーション実験結果 4

設定角度	90°				サンプリング周期：0.05秒
速 度 m/s	右 旋 回		左 旋 回		平 均 度
	走行のみ	脱穀部駆動	走行のみ	脱穀部駆動	
0.28	92.5	91.7	92.0	91.2	91.9
0.35	92.7	92.7	92.2	91.2	92.2
0.39	93.7	93.7	92.5	92.5	93.1
0.51	94.8	93.4	93.1	93.1	93.6
0.64	94.5	94.5	93.5	93.5	94.0
0.69	94.3	96.2	93.6	93.6	94.4
平 均	93.8	93.7	92.8	92.5	93.2

そこで各速度別の平均値を各実験毎にプロットした図を図3-37、38に示す。
「△」で表される理論値は図3-29、30で示したのと同じでありサンプリング周期0.2秒における理論値である。

45°の場合、実験とシミュレーションを比較すると低速度域で若干差はあるが変化の傾向としてはよく似ており、特に高速度域においてよい一致が見られる。理論的には設定値を下回る値はありえないため、低速度域においては旋回力が小さいため走行路面の影響を受けやすく、角速度が変化することによって積分誤差を生じることが原因であると考えられる。

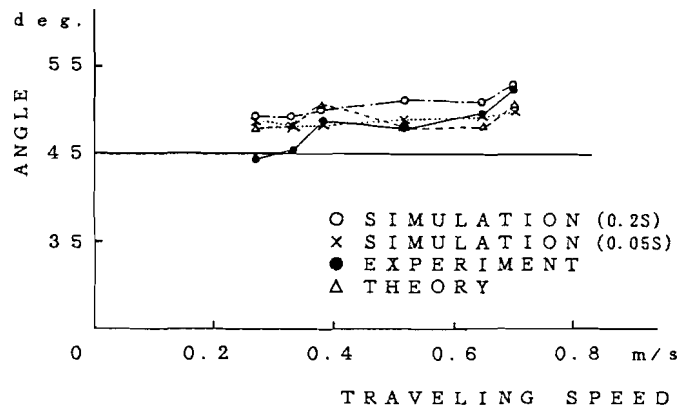


図 3-37 全実験結果の比較(45°)

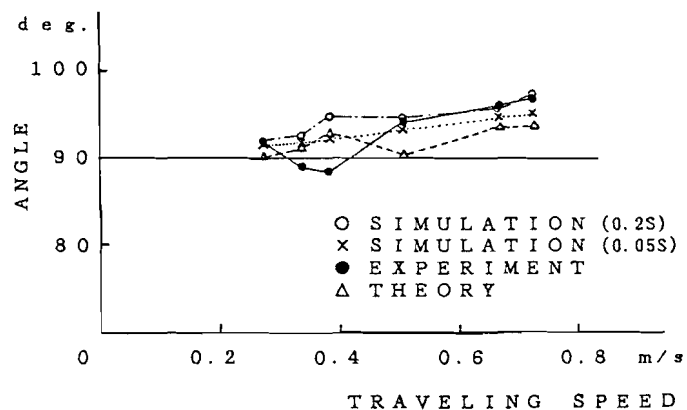


図 3-38 全実験結果の比較(90°)

90°の場合、低速度域の2点を除いて非常によく一致していると言える。またサンプリング周期を0.05秒の場合は、どちらの場合も速度によるバラツキが少なく、Ⅲ-4-(g)で考察したサンプリングしている時間による誤差が少なくなり、油圧装置の伝達遅れ時間による誤差が速度に比例して表われていることを示している。また安定度の点でも勝れていると言える。

以上のように、比較的高速度域における一致は見られたものの、低速度域における一致性を向上させるにはコンバインの走行モデルに走行路面の条件を付加する必要がある。

第 IV 章 回行制御

前章までに角速度センサを利用した旋回角検出装置の特性について考察してきたが、この装置をほ場末端部における回行制御に適用した。これは人間による回行手順をマイクロコンピュータのプログラムによって再現し、旋回角のフィードバック制御を加えることで高安定で高精度な制御を可能とした。本章ではこれらの制御方法について述べ、実験によって適用性を考察した。

IV-1 回行方法

一般的に人間による稲の刈取り方法はほ場端において90°旋回しほ場の周囲から順番に刈取っていく方法や180°旋回して反対側の条に進入し刈取っていく方法がある。後者は特にほ場が長方形の場合に利用される。また特殊な例として、上部旋回式コンバインではほ場端で180°旋回（上部のみ）して刈取った条の次の条に進入する研究^{76, 77)}が報告されている。本研究では最も一般的な90°旋回方法を自動化の対象とした。なお供試コンバインは全面刈りではあるが左回刈りが基本であるためこれに従った。

コンバインの走行制御をシステム的に行うには、直進区間の走行の自動化、及びほ場末端部での回行の完全な自動化が遂行されなければならない。回行の自動化は制御対象が不安定要素を含む土や作物であることや、回行時、制御対象と機体が離れるためにフィードバック制御が難しいなどの問題点があるが、一般に回行動作は図4-1の如き回行パターンで表示される。

- ① 刈取り作業中
- ② 条端を検出
- ③ 旋回のために一定時間前進し、刈取り部を上昇する。
- ④ 一定角度左旋回し、停止する。
- ⑤ 一定時間後進し、一定角度右旋回し、停止する。
- ⑥ 刈取り部を下降し、次の稲列に進入する。

このように回行動作は一連のシーケンス動作であるためプログラムによる制

御で容易に行えるが、それぞれの動作範囲をいかに規定するかがポイントである。そのための重要な要素として次のような事項が考えられる。

1. 条端検出後、旋回のためにしばらく前進する距離
2. 左旋回する角度
3. 後進する距離
4. 右旋回する角度

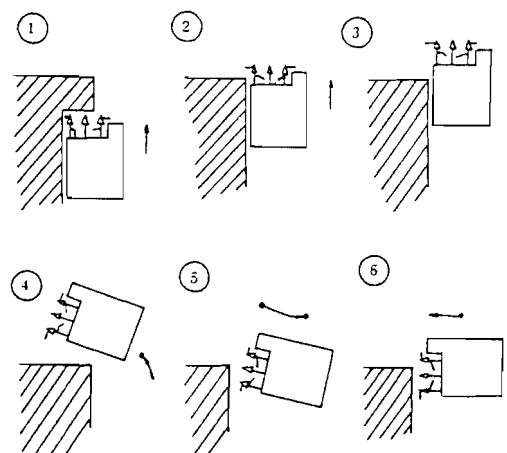


図 4-1 回行パターン

ここで2及び4については角速度センサによるフィードバック制御を行うが、1及び3については走行距離を直接検出せず、マイクロコンピュータのソフトウェアによるタイマにて時間設定し、距離に置換えて制御した。言うまでもなく走行部のすべりや速度の変化に対して無力であり、またほ場条件等により設定値を変更、調節しなければならないが、直進時におけるすべりはほ場条件にもよるがほとんど無視でき、十分対応可能であると言える。特に3については左外側センサによりコーナ部を検出するフィードバックループと時間ループとの2重ループにより確実性を向上させた。また少々すべりによるずれは回行終了後の操向センサによる補正で賄うことができる。

図4-2に補正の方法を示す。

- ①の如くコンバインが稲列に進入し過ぎた場合、右外側センサにより「進入過多」を検出し停止する。
- ②のように刈刃を上昇し、一定時間後退し、停止後刈刃を下降する。
- ③のように45°右旋回し、直進後
- ④のように45°左旋回し、機体を約35cm平行移動する。

この後稲列に進入し、それでもなお

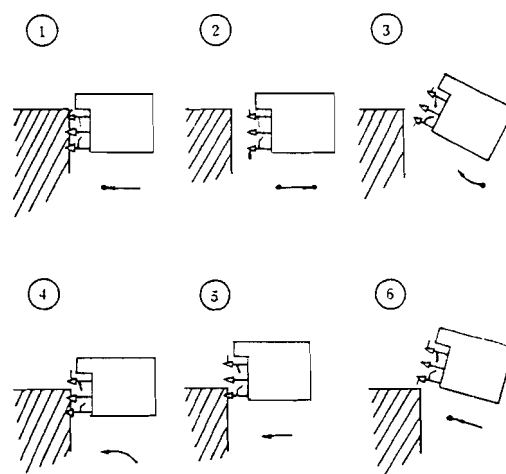


図 4-2 回行後の補正

右外側センサが稲を検出した場合、このセンサが稲を検出しなくなるまでこのシーケンスを繰り返す。

また⑤⑥の場合のように右操向センサが一定時間内に稲を検出しなかった場合はコンバインは停止する。

IV-2 制御プログラム

図4-3に回行プログラムの概略フローチャートを示す。本プログラムはサブルーチン形式を取り自動走行プログラムにより条端を検出し、一定距離前進後、本プログラムがコールされ回行が実行される。

最初に旋回による切株への接触を防ぐため刈取り部を上昇し、サブCPU部をリセットして

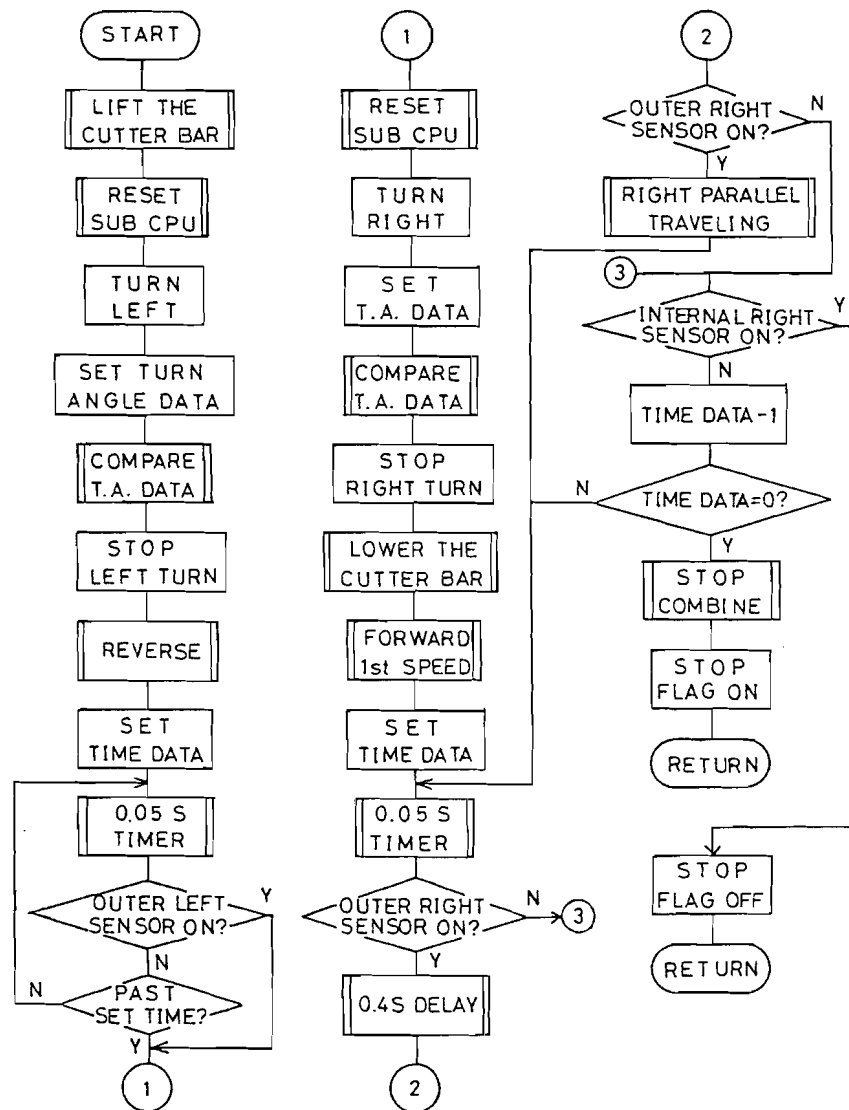


図 4-3 回行プログラムの概略フローチャート

左旋回する。この間サブCPU部から現在の旋回角をフィードバックしこれが設定値を越えると旋回を終了し、予め設定された時間に達するかまたは左外側のセンサがコーナ部の稲を検出するまで後進する。そしてサブCPUをリセッ

トし設定角度に達するまで右旋回し刈取り部を下降して、停止する。このようにして回行を終了し次の稲列に進入する。この時、右外側センサが稲を検出した場合、0.4秒後に再度検出を確認して35cm右に平行移動する。また一定時間以内に右操向センサが稲を検出しなければ、刈取り終了と判断し、停止フラグをONにして、稲を検出すれば停止フラグをOFFにしてリターンする。

IV-3 実験方法

実験は実際に通常自動走行プログラムによって稲の刈取り作業を実施し、コーナの形状を変化させて、コンバインの走行軌跡から回行制御の適応性を調べた。走行速度は約0.5m/sとした。また左旋回角は55°右旋回角は25°に設定した。この値は前章で考察したように実際の旋回角は設定値よりも大きくなるために実際の走行試験から小さめの値に設定した。コンバインの走行軌跡は左右のディバイダ先端としたが、実際にはクローラによる消失を防ぐためコンバイン右側約1mにマーカとして白色に着色した流体ゲルを滴下した。軌跡の測定はほ場を座標平面として考え、適当に基準線を決定してマーカの位置のx座標、y座標を実測した。また稲の位置は刈取った切株から測定した。このようにして実測した値はディバイダ先端の軌跡に修正しなければならないが、コンバインの回転中心や半径がたびたび変化するためすべての実測値を同一式のもとに置換えることは困難であるため、実測値をプロットして得られた軌跡上にコンバインのマーカの位置を重ねつつ、コンバインの図形を回転させ、最も適当と思われる点を回転中心とみなして修正を行った。

IV-4 実験結果及び考察

実験は8回行い、それぞれコーナ部の形状や角度を変化させ、縦刈りから出発し、回行後、横刈りで終了し、走行軌跡と切株の位置を記録した。これをディバイダ先端の軌跡に修正し、株の位置とともにプロットした。この結果を図4-4、図4-5に示す。また流体ゲルによる実際の走行軌跡を図4-6に示す。また走行軌跡から回行動作を分割して構成要素別にまとめた表を表4-1に、そして

これらの統計解析結果を表4-2に示す。表中のそれぞれの時間は走行軌跡から得られる距離を速度で割った値である。なお実験番号5、6が横刈り走行時において稲列に追従しなかった。

ここで「後進時間2」は右旋回終了後、停止するまでの時間を、また「交差角」は条端検出後のコンバインの脱出方向と次の稲列の進入方向とのなす角を、「COM」は設定値をそして誤差率は設定値に対する平均値の百分率を表している。

「前進時間」は最後の稲の位置と検出タイミングにより異なるためどうしてもばらつきが大きくなる傾向があり、変動係数も大きくなっている。

「左旋回角」は角速度センサによるフィードバック制御によるため設定値よりも大きくなるはずであるが実際の値は小さくなっている。これは左操向用シリンドラの不調により、完全に履帯を制動することが不可能であったため、人間による補助を加えたことが原因となったと考えられる。

「後進時間」はどちらも比較的安定した値となっており、直進に関しては時間による制御で十分可能であることが確認された。

各実験区のコーナの形状は「鋭角」、「直角」、「鈍角」に大きく分類されるが稲の配置が離散的であるために細かく角度を設定することが不可能であったため、株を適当に間引くことで形状を変化させた。

実験番号1は2条のうち左側の稲列が突出しており、このため前進時間が長くなり進入不足になると予想したが、前進時間は設定より短く、非常に良い結果が得られた。これは突出部の稲が機体との位置関係により検出されなかったためと考えられる。

実験番号2はコーナの角度はやや鈍角であり条列の位置以外に余分の株が存在するがまったく問題なく走行している。また交差角が若干小さかったため横刈り走行時に修正されているのが確認できる。

実験番号3はコーナの角度はほぼ直角であるが交差角が実験区の中で最小であるため修正が若干遅れぎみであるが徐々に修正されていることがわかる。これは横刈り時には追従の時定数を大きくしてあることが原因である。

実験番号4は横刈り進入時のコーナ形状を一株間引くことにより鈍角にしているため稲の検出が遅れていることやその後、鋭角になっているため追従がやや苦しかった。

実験番号5はコーナ角度が鋭角であるため前進時間が長くなったことと交差角が小さいため条件的に最も不利となったため追従不可能であった。

実験番号6はコーナ角度は鋭角であるが横刈り時の稲の位置を大きく変化させその追従性を調べた。まず前進時間が最も長く交差角も若干小さいことから厳しい条件となり、追従不可能であった。前進時間が長いのはセンサが条端を通過した後に切株を検出したかワラ等が絡まったためと考えられる。

実験番号7はコーナ角度は鋭角であるが所々株を間引いてあるが、回行が比較的順調に終了し、横刈り時の追従も良好であった。

実験番号8はコーナ角度が鋭角で突出株が2株とかなり厳しい条件であったが交差角が大きいため進入は良好であったが、センサが稲から離れ左へ修正したため、右側の株をはずしていることがわかる。

回行の成否はそのパターンがある程度くずれても最終的には右操向センサが稲列を検出し、追従すれば成功と考えてよい。これは回行後のコンバインの進行方向と稲列との位置関係によって決定される。そこで回行後のコンバインの方向と稲列の方向との相対角度と最初の稲とディバイダとの垂直距離との関係を調べグラフ化した。これを図4-7に示す。

ディバイダと稲との垂直距離が右操向センサとの位置関係から30cm以内であれば稲の検出が可能となるが、コンバインと稲列との相対角度が大きくなれば次の稲に対する修正角が大きくなり追従不可能になる場合がある。グラフから実験番号5、6は明らかに30cmを越えているため追従しなかったことがわかる。また実験番号3は $\alpha = 15^\circ$ と大きいが $l = 15\text{cm}$ であったため追従可能であった。実験番号4はこの逆の関係であるが同様のことが言える。両者とも追従は遅れぎみであったことからこれらの数値が本制御の適用限界であると言える。つまり $-15^\circ \leq \alpha \leq 15^\circ$ かつ $l \leq 30\text{cm}$ であれば追従可能であると考えられる。

この限界を拡げるためには横刈り走行におけるコンバイン追従の時定数を小

小さくすることで追従性を向上させることが可能であるが、横刈り時は株間が広い
ためハンチングを生じやすい。このためハンチングを生じない範囲でなるべく
時定数を小さくすることが追従性の向上に有効である。

表 4-1 回行動作の構成要素の諸数値

実験番号	前進時間 (秒)	左旋回角 (度)	後進時間 1 (秒)	右旋回角 (度)	後進時間 2 (秒)	交 差 角 (度)
1	2.9	58.0	3.9	33.0	1.5	91.0
2	3.3	53.0	3.9	33.0	1.3	86.0
3	3.3	47.0	4.1	27.0	1.4	74.0
4	3.4	52.0	4.0	31.0	1.4	83.0
5	3.9	52.0	3.9	27.0	1.5	79.0
6	4.5	52.0	4.1	30.0	1.6	82.0
7	4.2	57.0	4.2	32.0	1.4	89.0
8	4.3	55.0	4.0	40.0	1.3	95.0
COM	3.3	55.0	4.0	25.0	1.6	80.0

表 4-2 回行構成要素の統計解析結果

項 目	最大値	最小値	平 均	標準偏差	変動係数%	誤差率%
前進時間 (秒)	4.5	2.9	3.7	0.58	15.5	12.1
左旋回角 (度)	58.0	47.0	53.3	3.45	6.5	-3.1
後進時間 1 (秒)	4.2	3.9	4.0	0.11	2.8	0.0
右旋回角 (度)	40.0	27.0	31.6	4.14	13.1	26.4
後進時間 2 (秒)	1.6	1.3	1.4	0.10	7.3	-12.5
交 差 角 (度)	95.0	74.0	84.9	6.79	8.0	6.1

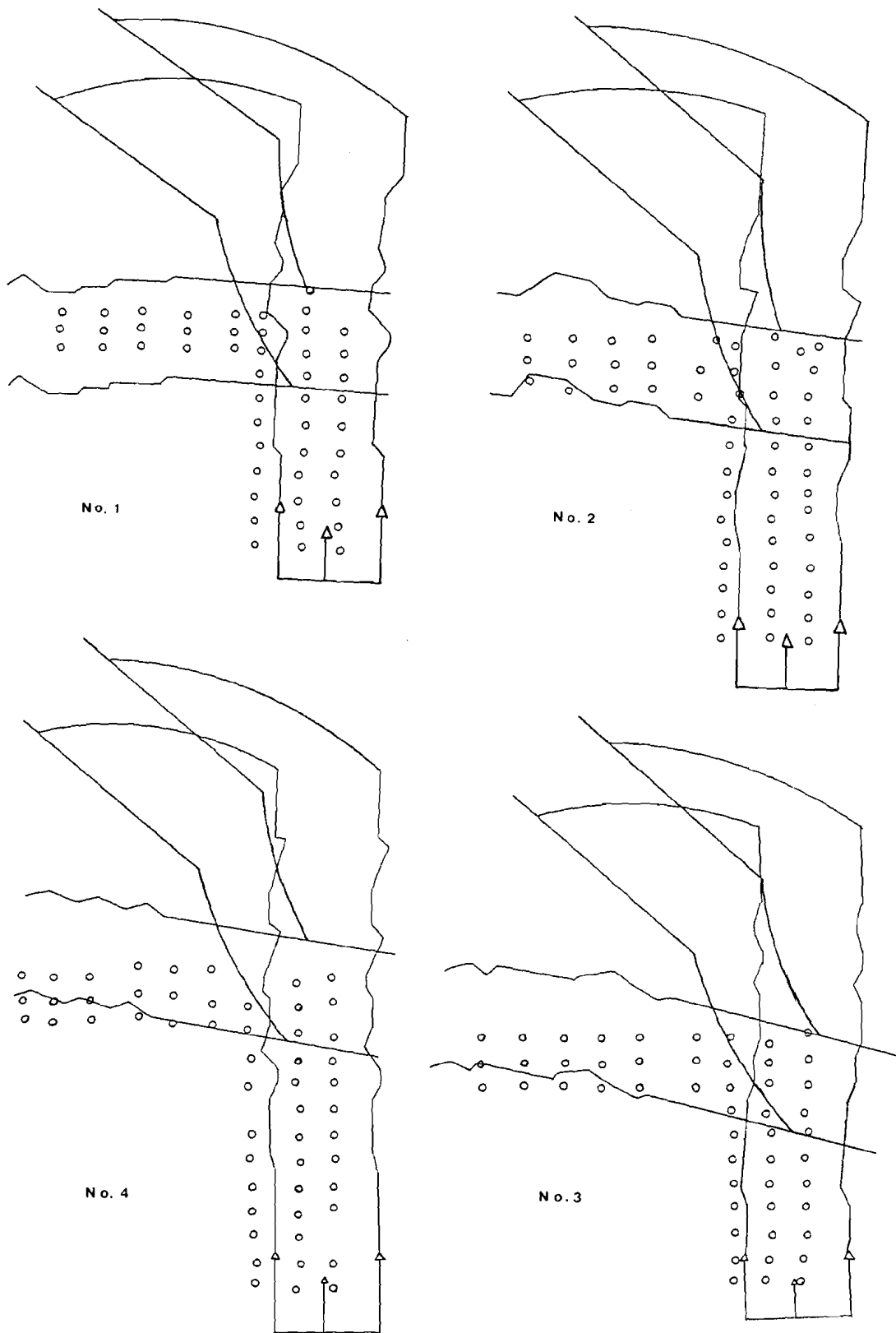


図 4-4 回行時の走行軌跡 1

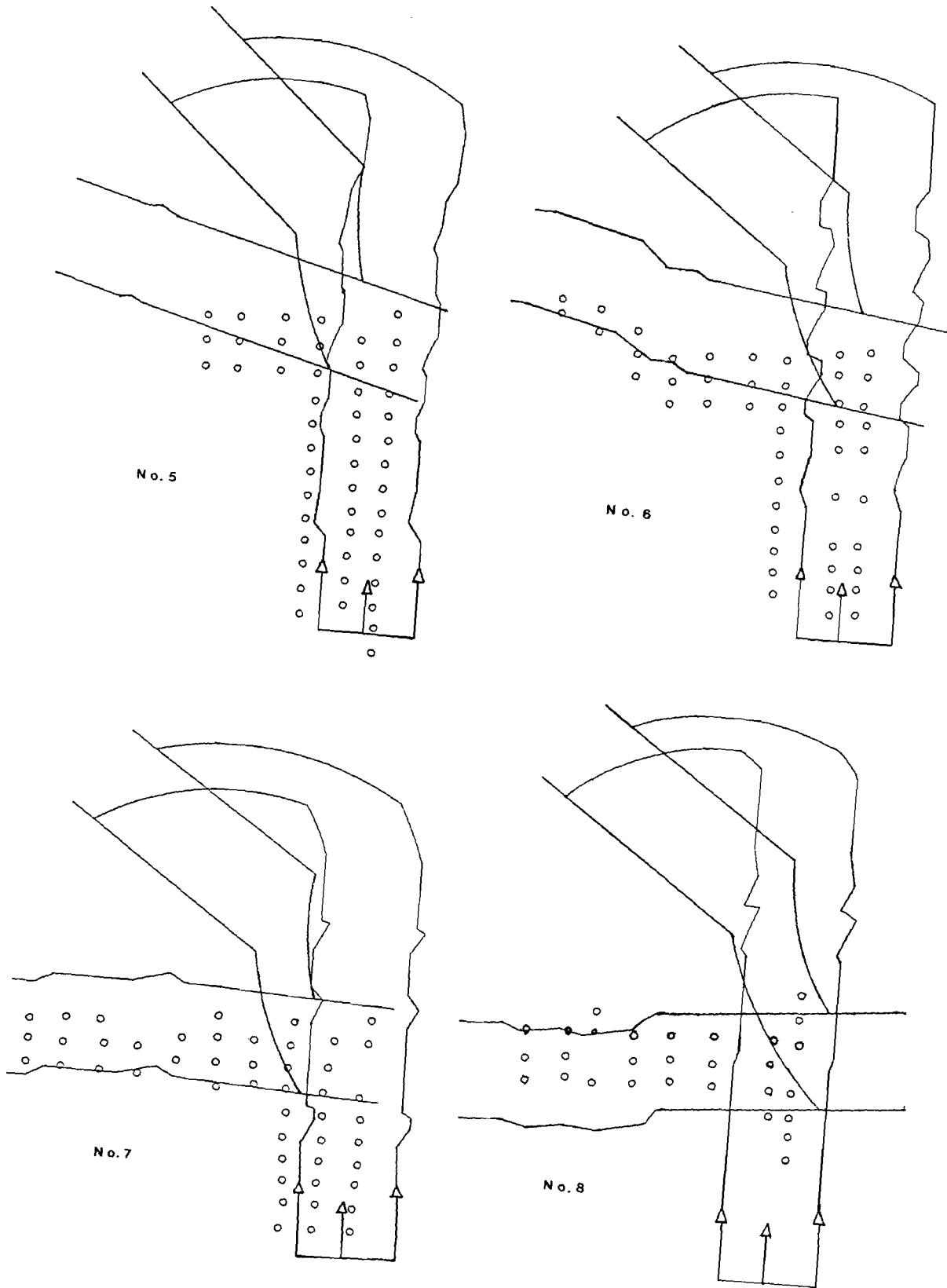


図 4-5 回行時の走行軌跡 2



図 4-6 流体ゲルによる走行軌跡

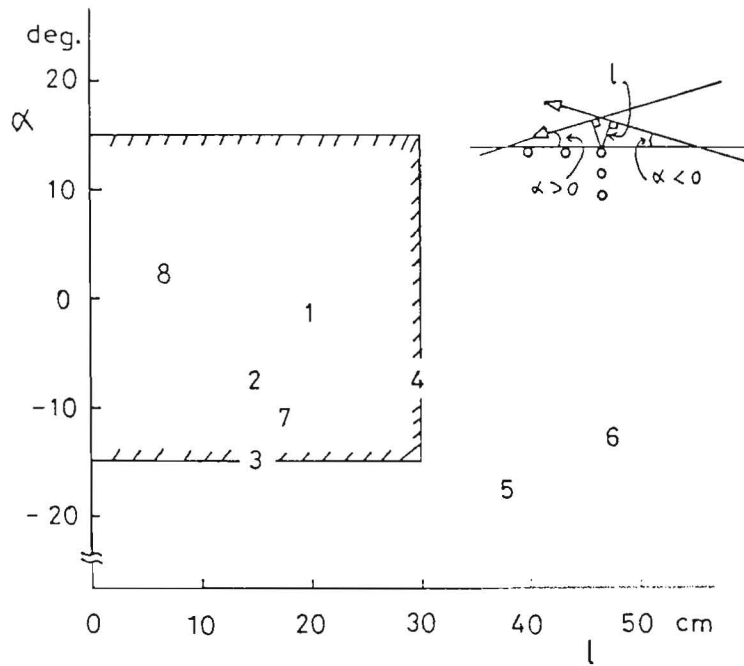


図 4-7 回行成否の限界領域

第 V 章 最短距離制御

前章では旋回角検出装置の適用例として回行制御について論じたが、本章ではもう一つの適用例として「最短距離制御」と称する刈取り作業時における作業能率の向上を目的とした制御方法を考案し、実際の刈取り作業への適用性について基礎実験を通じて考察した。

V-1 最短距離制御の概念

最短距離制御の概念の大略については第 I 章で既に述べたが、この制御が可能となる理由は以下に述べる 2 点に集約される。

第 1 点目は自脱コンバインの刈幅と稲の条間との関係から生じる刈取り余裕の存在である。

通常、自脱コンバインの刈取り条数は 2～6 条であり、供試コンバインは 2 条刈でその刈幅は 77 cm である。図 5-1 にコン

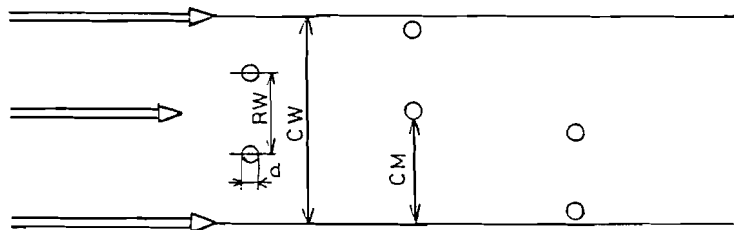


図 5-1 ディバイダと稲との位置関係

バインのディバイダに対する稲との位置関係を示す。稲はディバイダ内に存在すれば刈取り可能となり、条間に対して刈幅に余裕が生じる。この余裕 CM は式 (5-1) で表すことができる。

$$CM = CW - (RW + d) \quad (5-1)$$

ここで、 CW : 刈幅、 RW : 条間、 d : 株の直径

今、 $CW = 77\text{cm}$ 、 $RW = 30\text{cm}$ 、 $d = 6.6\text{cm}$ とすれば、 $CM \approx 40\text{cm}$ となり、これは稲列の曲がりの最大振幅が 20cm 以下であれば進行方向を適切に選択することによって、全く操舵することなく直進走行が可能となることを意味している。

近年、田植機における植付け条数が複数条化しており最大12条植えの田植機も出現している⁷⁸⁾。これは同じような曲がりを持つ稲列が植付け条数分存在することになる。これが第2点目の理由である。つまり最も最適な進行方向を決定するために、最初の列を利用して刈取り作業を実行しつつ稲の曲がりのデータを収集し、2回目以降はそのデータを基にして最適な進行方向（最短距離方向）を計算して刈取り作業を実行することが可能となる。具体的に12条植えの田植機で移植し、2条刈のコンバインで収穫する場合、最初の2条で稲の曲がりのデータを収集し、その後5回、そのデータによる最短距離走行が可能となり、次の稲列では新たにデータを収集し、最短距離走行を行うことが繰り返される。このため田植機は偶数条植えが前提となり、多条植えになるほど本制御は有効となる。また、作業開始前に植付け条数を入力する必要がある。但し横刈りでは稲列を形成していないため稲列データの収集は行わないものとする。

V-2 制御方法

最短距離制御の概念を具体化するためには稲の曲がりのデータの収集方法とこのデータから最短距離を演算する方法の確立が必要となる。本項では旋回角検出装置を利用したこれらの制御方法について述べる。

V-2-(a) 稲列データの収集方法

コンバイン自身が実際の稲の曲がりそのものをデータとして収集することは非常に困難であるため、コンバインが稲列に沿って走行した状態を稲列の曲がりのデータとして代表した。コンバインの走行状態は直進動作と旋回動作のどちらかであり、操向センサが稲を検出している間は旋回動作を行い、それ以外は直進動作と考えることができる。つまり旋回動作は旋回角検出装置によってその旋回角を、直進動作はマイクロコンピュータによるタイマによってその時間を計測することでコンバインの位置を確定することが可能となる。この具体的方法を図5-2に示す。

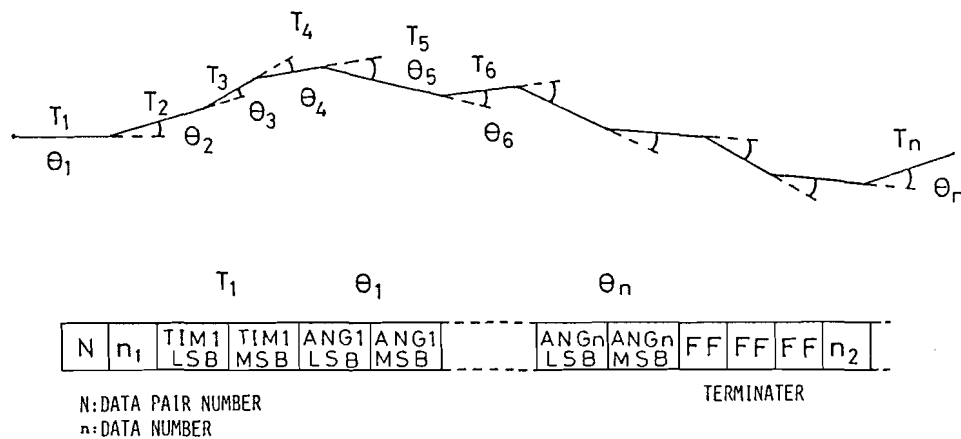


図 5-2 稲列データの収集方法

コンバインは左右のディバイダ内側に設置したセンサによって稲列に追従し、操舵間の時間を0.1秒単位でまた操舵角を旋回角検出装置によって計測し、これらをマイクロコンピュータのメモリに記憶して稲列のデータとした。時間計測は割込みを利用して正確に計測した。また図中下部にメモリに記憶されるデータのフォーマットを示す。最初の1バイト目にデータの組数、2バイト目に一組のデータに含まれるバイト数、また時間及び操舵角データはそれぞれ2バイトのBCDで表され、下位バイト、上位バイトの順に格納される。そして最後に区切記号のFFHが3バイト格納される。

V-2-(b) 制御プログラム

稲列データ収集の概略フローチャートを図5-3に示す。

プログラムは時間計測のタイマをスタートさせ、すべてのセンサの状態を入力し、稲を検出したならば左右のセンサの状態を入力する。仮に左側のセンサがONであればそれまでの時間を演算し、センサがOFFになるまでの操舵角をサブCPU部より入力して、それぞれの値をメモリに書き込む。これを稲が検出されなくなるまで繰り返し、稲列データの収集を終了する。また操舵力は回行時に適合するように最大に設定されているため、ハンチング現象を生じたり稲列をはずすことがある。この対策としてソレノイドバルブをパルス駆動し、

平均パワーを下げることで操舵力を加減した。

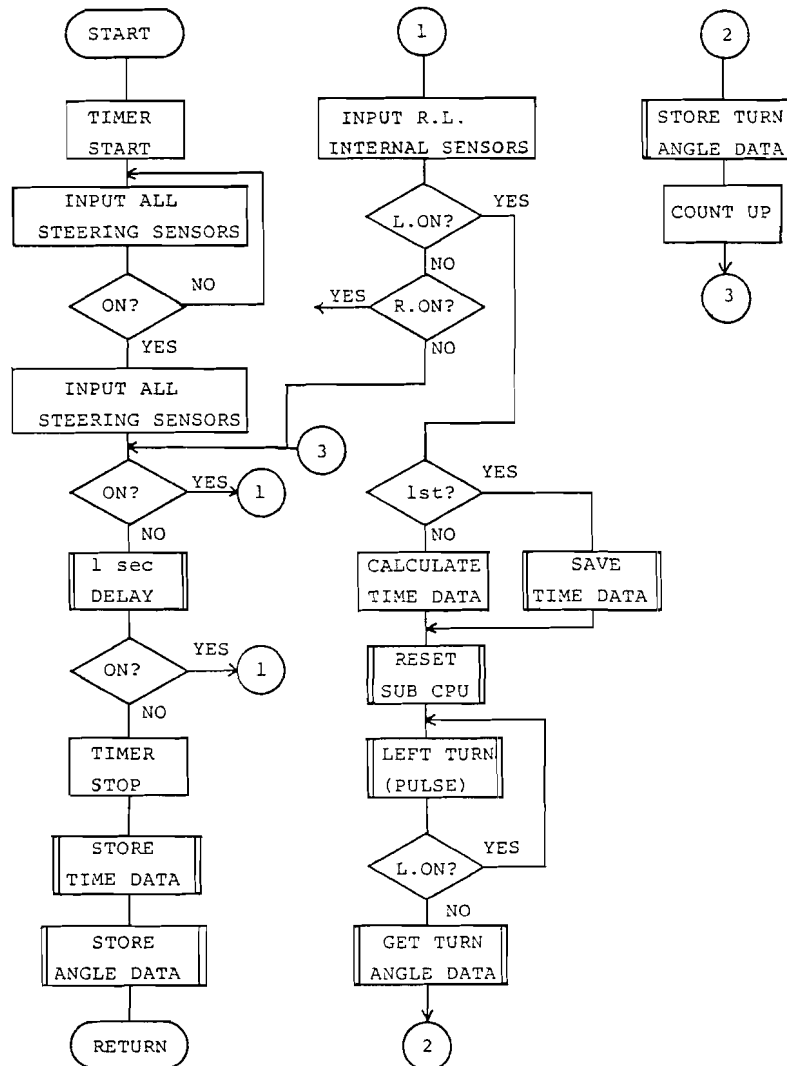


図 5-3 稷列データ収集プログラムの概略フローチャート

V-2-(c) 最短距離演算方法

収集された稷列データから最短距離方向を決定しなければならない。この目標点をどのようにして選択するかが大きなポイントとなる。稷列データは左右の操舵の繰り返しにより、極値が存在することになり、この点を目標点とした場合にその点までの各点が刈幅内に存在することが必要条件となる。しかしながら、この演算は非常に困難であるため、各操舵点とその前後の位置関係によって目標点となるかどうかを判断した。

稲列データは時間と角度のデータの一種の極座標系であるため、極値を求めるために直交座標系に展開した。図5-2において進行方向をx、その直角方向をyとした場合、任意の点nにおける座標は式(5-2)で表される。

$$\begin{aligned} x_n &= V \cdot T_n \cdot \cos \sum_{k=1}^n \theta_k + x_{n-1} \\ y_n &= V \cdot T_n \cdot \sin \sum_{k=1}^n \theta_k + y_{n-1} \end{aligned} \quad (5-2)$$

ここでV：走行速度、 θ ：右旋回を正

このようにして展開された座標から極値を抽出するための条件を式(5-3)で定義する。

$$\begin{aligned} y_n > y_{n+1} \cap y_n > y_{n-1} \cap |y_n - y_{n+1}| > k \cap |y_n - y_{n-1}| > k \\ y_n < y_{n+1} \cap y_n < y_{n-1} \cap |y_n - y_{n+1}| > k \cap |y_n - y_{n-1}| > k \end{aligned} \quad (5-3)$$

これは任意の点のy座標がその前後の点のy座標よりも大きいかまたは小さくかつその座標の差の絶対値がある値kよりも大きいことが極値となる条件である。このkの値を適当に選択することによって、目標点を変えることが可能となる。このようにして抽出された座標点はコンバインの走行のデータである角度と時間のデータに逆変換する必要がある。目標点の座標を (x_{0n}, y_{0n}) とした場合、その点までの時間と距離は式(5-4)で求められる。

$$\begin{aligned} \therefore T_{0n} &= \sqrt{(x_{0n} - x_{0n-1})^2 + (y_{0n} - y_{0n-1})^2} / V \\ \therefore \theta_{0n} &= \tan^{-1} (y_{0n} - y_{0n-1}) / (x_{0n} - x_{0n-1}) - \theta_{0n-1} \end{aligned} \quad (5-4)$$

以上の演算は演算時間短縮のため2倍精度の固定小数点演算法^{79, 80)}で行った。sin及び \tan^{-1} は線形近似、またcosはテーブル参照法で行ったため $\pm 25^\circ$ を越えると誤差が大きくなるが、実際の適用範囲から考えればそれほど問題にはならない。

V - 2 - (d) 制御プログラム

最短距離走行プログラムの概略フローチャートを図5-4に示す。

プログラムはまず収集した稲列データから直交座標変換、極値の抽出、目標点までの時間と角度の計算を行い、第1目標点までの時間を計測しつつ直進し、到達後、第2目標点までの角度データを取り出しそのデータ分操舵し、次の目標点までの時間を計測する。これをデータがFFFFHになるまで繰り返す。

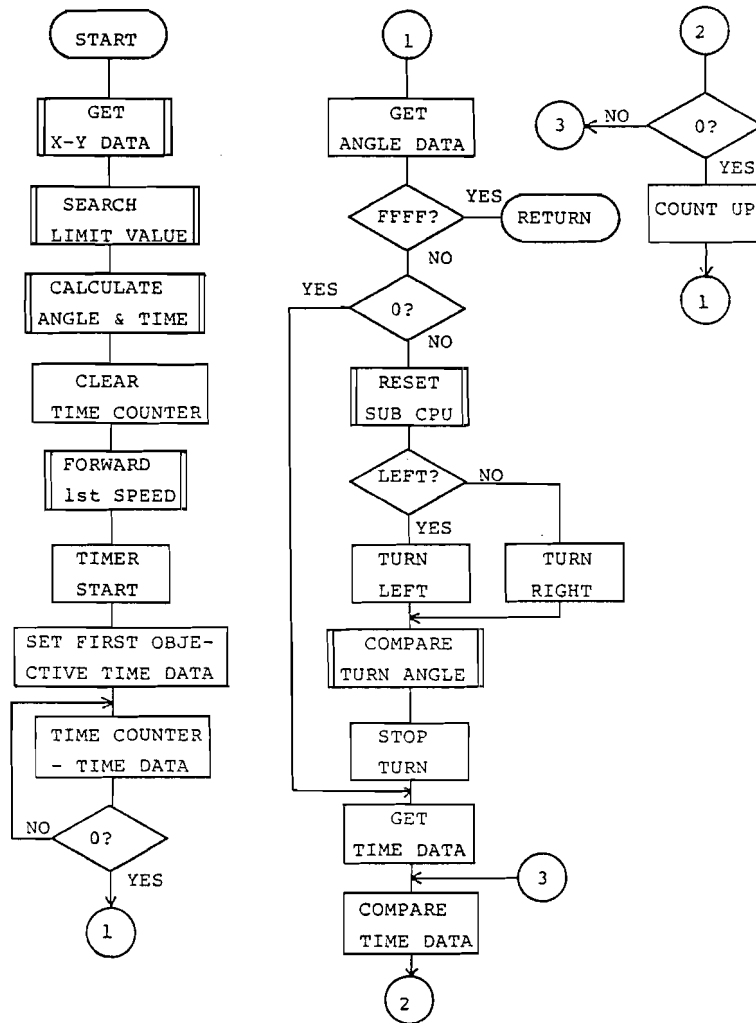


図 5-4 最短距離走行プログラムの概略フローチャート

V - 3 実験方法

最短距離制御の実行を確認するためにほ場において実験を行った。これは稲の刈取りは行わず、コンバインをセンサの代りに接続したスイッチによる遠隔操作で左右の操舵を繰り返し、十数mにわたって走行することで稲列データ収集を再現した。そしてクローラの中心で代表した走行軌跡を20cm間隔で測定した。次に収集したデータによる最短距離走行を実行し、その走行軌跡を同様に測定した。これらの走行軌跡を図5-5、5-6に示す。



図 5-5 稲列データ収集走行軌跡

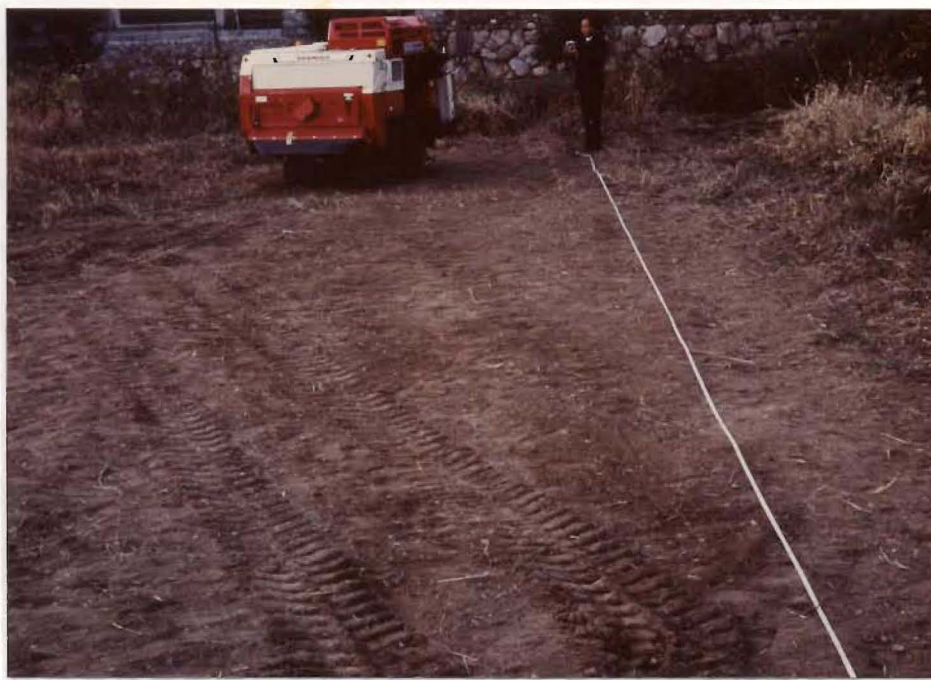


図 5-6 最短距離走行軌跡

また稲列データは実験終了後に搭載マイクロコンピュータのメモリからI/Oポートを経由して外部コンピュータ(PC8801)に転送し、同一演算ルーチンで最短距離演算を再現した。図5-7に転送された稲列データの一例を示す。

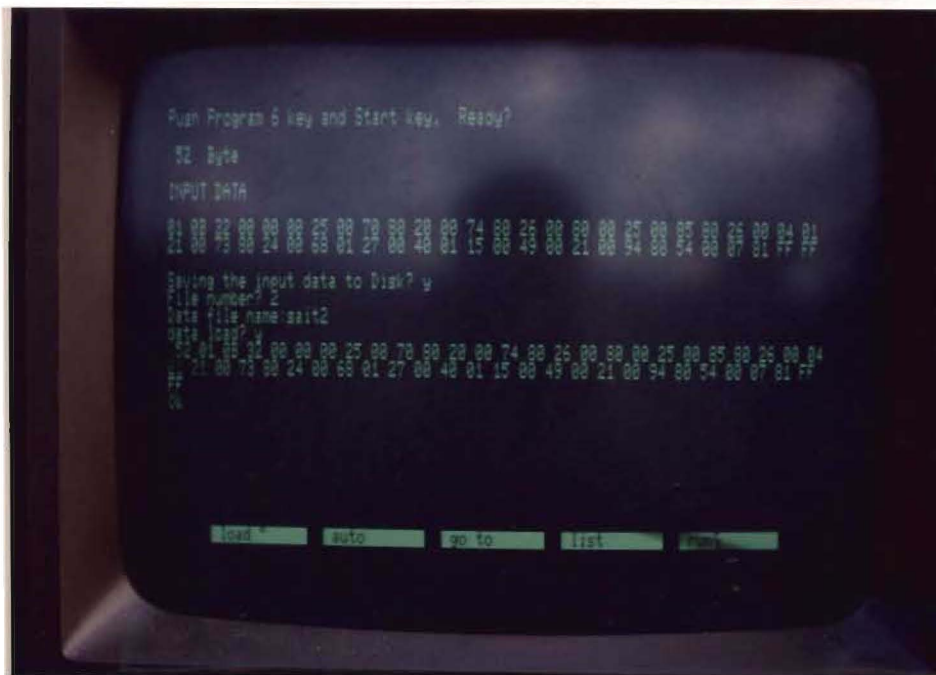


図 5-7 転送された稲列データ

V-4 実験結果及び考察

実験は4回行い、走行速度は0.5m/sに設定し、極値判定の条件であるkの値は距離に換算して約1mとした。これらの走行軌跡を図5-8~11に示す。各図の中で上の図は稲列データより演算した仮想走行軌跡、下の図が実際の走行軌跡を表している。

各実験区における最短距離演算は非常に正確な演算が行われており、三角関数等の近似計算による誤差の影響はほとんどないことが確認された。次に収集した稲列データを速度から距離に展開した仮想走行軌跡と実際の走行軌跡との偏差を計算し、その再現性を調べた。これを図5-12に示す。

どの実験区においても距離が長くなるほど偏差が大きくなる傾向を示している。これは旋回角検出時における誤差が距離が長くなるほど累積されるためと考えられるが、パルス操舵により角速度の変化がステップ状になり、操舵時間

が長くなったため比較的誤差が小さくなり、全体として良好に再現されていると言える。

また仮想走行軌跡と実走行軌跡との形状の類似性を調べるために両軌跡間の相関係数と平均自乗偏差の根である rms 値を求めた。これを図5-13に示す。縦軸に相関係数、横軸に rms 値をとりアンダーバーのある記号が最短距離走行、ないのが稲列データ収集走行を表している。実験番号1及び3は rms 値が低くて相関係数が高く、比較的良好的な結果と言える。実験番号4に関しては稲列データ収集走行の再現性は良好であるが最短距離走行の rms 値が50cm近くありズレが大きいことがわかる。

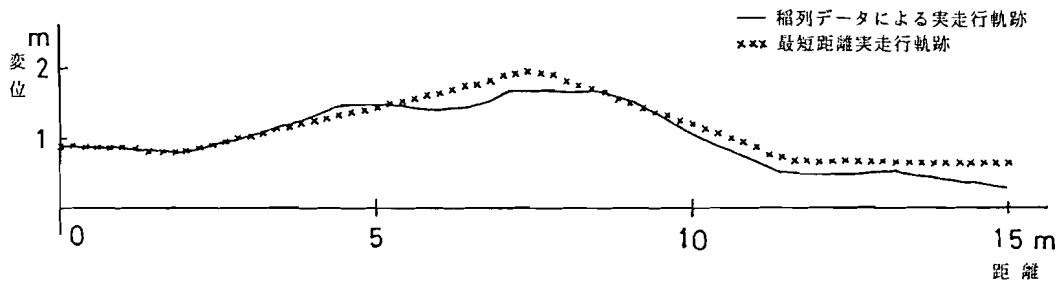
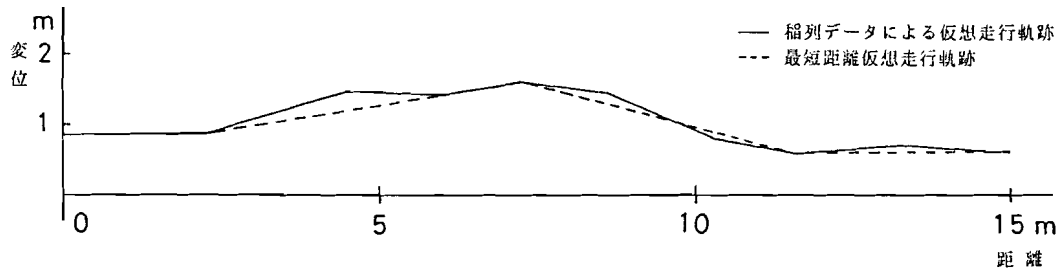
稲列データによる実走行軌跡から最短距離演算によって抽出された目標点に対する最短距離実走行軌跡における目標点（到達点）との偏差を示した図を図5-14に示す。

矢印は目標点の順序を表しているが、第1目標点はほとんど偏差が0であったため図では省略した。

実験番号1はすべての目標点が半径20cm強の円内に収まっており、非常に良い結果と言える。

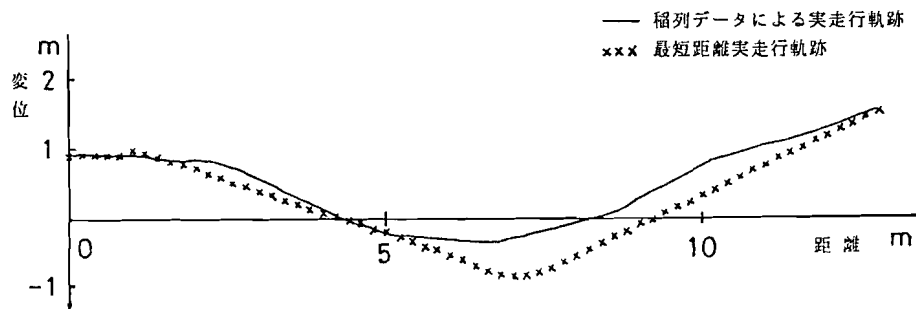
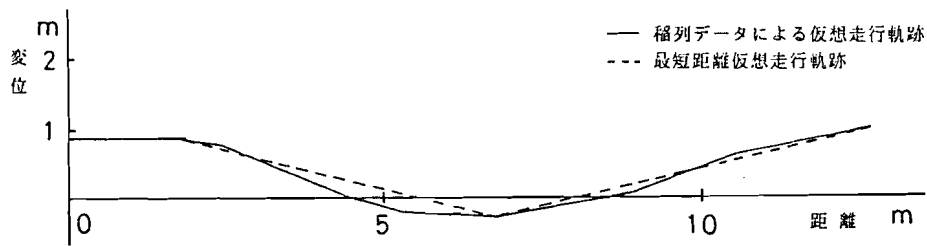
他の実験区においては第2目標点での偏差が大きく、最終目標点での偏差が非常に小さいという傾向が見られる。通常、距離が長くなるほど誤差が累積され偏差が拡大するが、この実験のように左右の極値が繰り返される場合には左右の誤差が打ち消しあい結果的に誤差が小さくなったと考えることができる。

以上のように目標点における偏差の平均は半径30cm以内に収まり、比較的良い結果と言えるが、目標点の違いによる偏差の差が大きい。これは第III章で考察したように旋回角の設定値に対して実際の旋回角が必ず大きくなることに起因している。このため演算して求められた値より小さな設定値となるようにプログラムを変更することにより精度向上を計ることが可能となる。今後、実際の稲列への適用には操向センサによる補正を加えることで対応可能と考える。



No. 1

図 5-8 走行軌跡 1



No. 2

図 5-9 走行軌跡 2

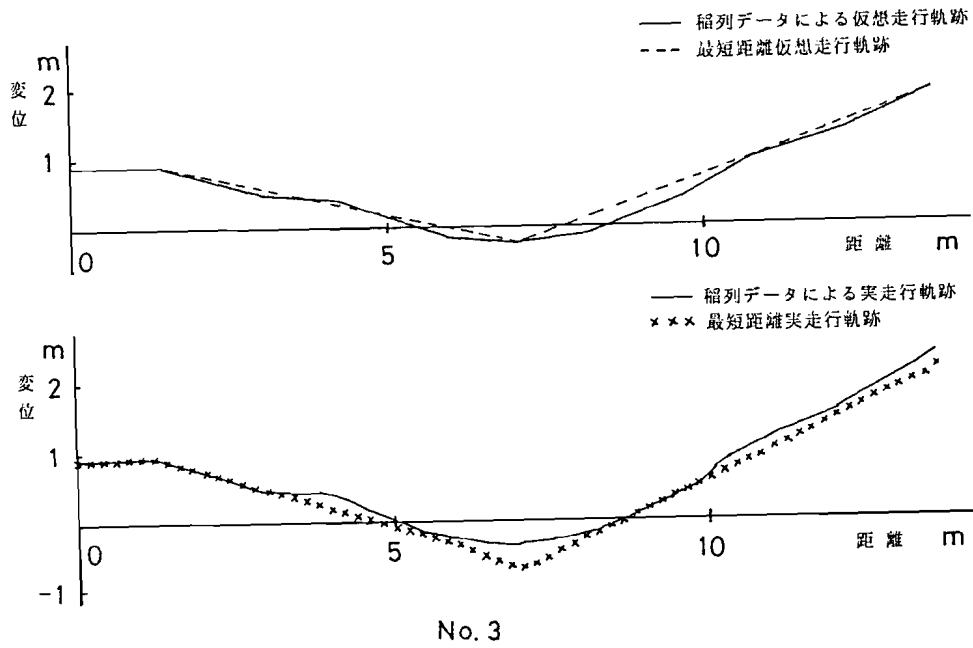


図 5-10 走行軌跡 3

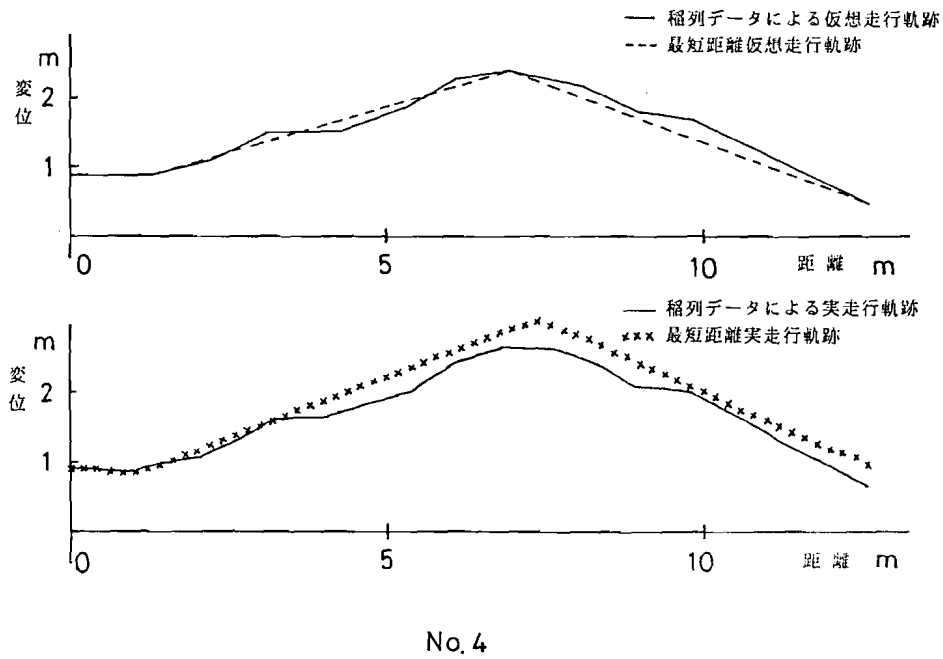


図 5-11 走行軌跡 4

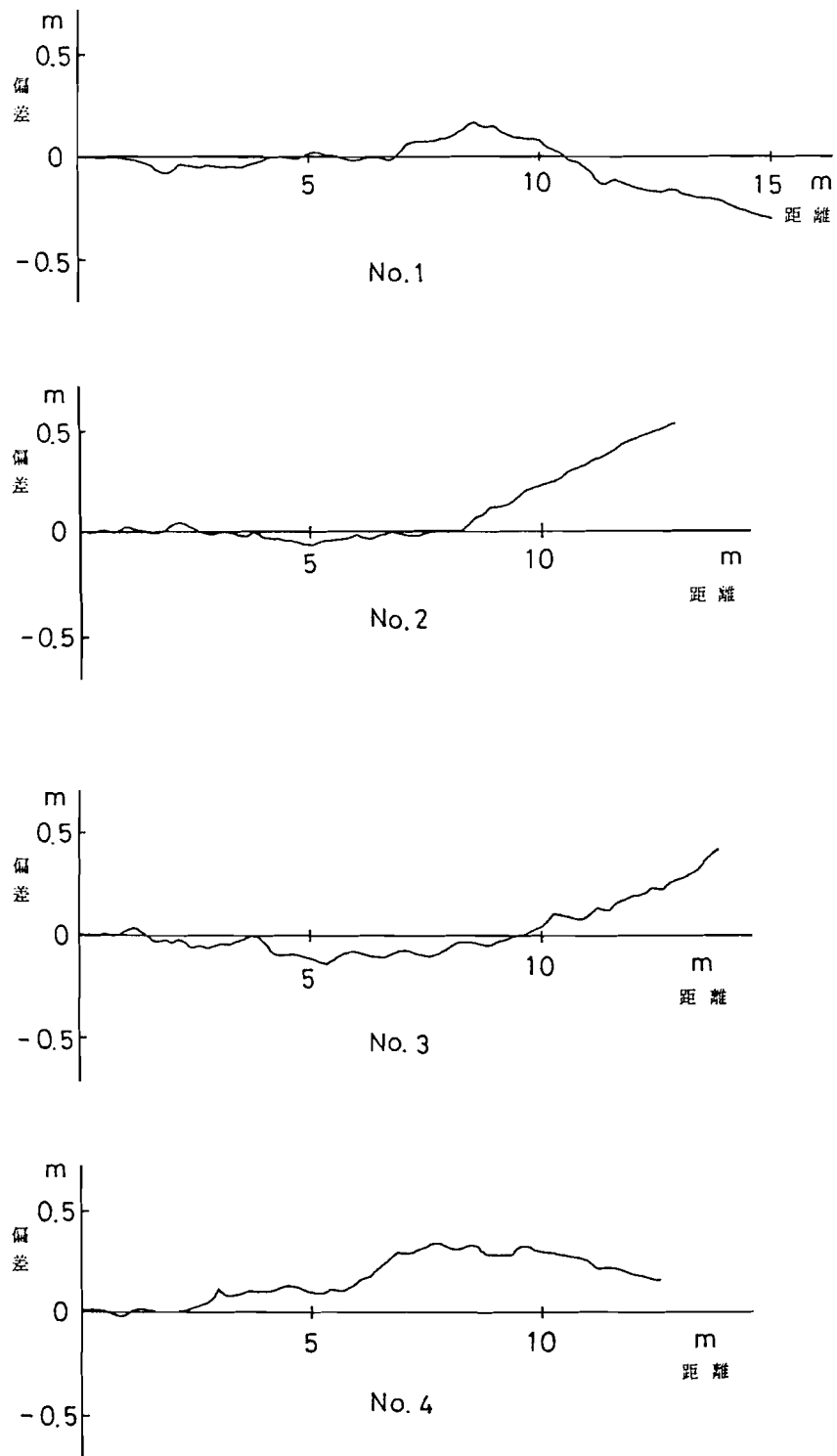


図 5-12 稲列データによる仮想走行軌跡と実走行軌跡との偏差

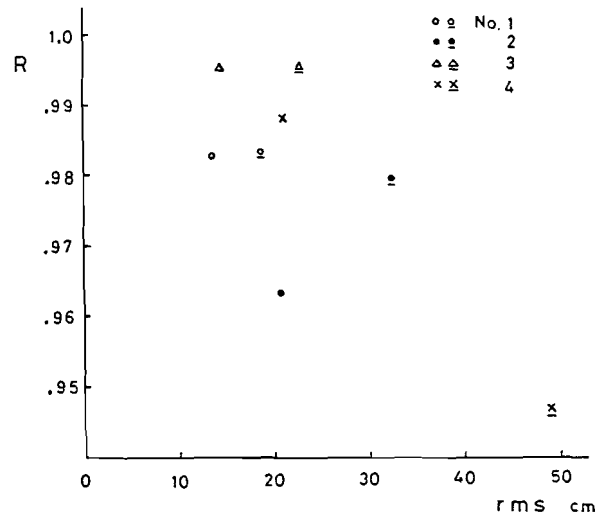


図 5-13 仮想走行軌跡と実走行軌跡間の相関係数と rms

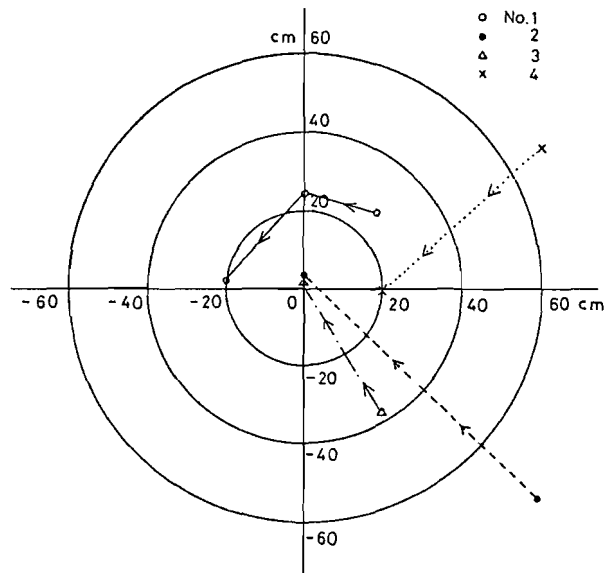


図 5-14 目標点における偏差

摘 要

自脱コンバインの自動化のうち、特に移動機械としてのコンバインに注目し、高精度・高安定な走行を実現するためにコンバインの旋回角を非接触で検出する方法を考案した。そしてこの方法を応用したほ場末端部における回行制御及び刈取り作業時の能率向上を目的とした最短距離制御について論じた。

第Ⅰ章では、ほ場走行機械の自動化に関する研究を外観したのち、回行制御の問題点について明らかにし、旋回角検出の必要性について述べた。コンバインは回行時に稲から離れるために稲との相対位置の検出は非常に困難である。そこで旋回角を角速度センサとマイクロコンピュータによる旋回角検出装置によってフィードバックし、回行シーケンスを実行することで走行部でのスリップによる影響を減らすことが可能となった。

現在、田植機による移植が盛んに行われ植付け条数も複数条化しており、オペレータの熟練度にもよるが一般的に稲列の曲がりを伴うことが多い。稲の曲がりに関する調査の結果、約40m間に25cmp-pの振幅を持つことが判明した。また地上高8cmにおける株の直径は標本数677の平均で約7cmであった。本研究では2条刈りの自脱コンバインを供試するが、この刈幅は約77cmであり、稲の条間を30cmとすれば株間は37cmとなり40cmの刈取り余裕があると言える。これは目標を適切に選択し、稲列の曲がりの最大振幅が20cmp-p/条以下であれば、全く操舵することなく、直進走行が可能であり能率向上に役立つと言える。しかしながら、目標を適切に選択することは不可能に近く現実的ではない。そこで2条刈りのコンバインと複数条植え（偶数条）の田植機に注目し、最初の刈取り作業で稲の曲がりのデータを収集記憶し、2回目以降はそのデータを基に最短距離を計算して、刈取り作業を実施する制御をしようとするものであり、多条植えになるほど有効である。但し、横刈り走行では稲列が形成されていないためデータの収集は行わない。この制御を最短距離制御と呼ぶ。

第Ⅱ章では、自動制御のシステム設計に先立ち、制御対象となる稲の状態を把握するために、稲の生育状態及び稲列の曲がりを調査を行い、その結果につ

いて述べた。

調査項目は稲の地上高 8cmにおける株の直径、地上からの伸長時の草丈及び稲列の曲がりである。これらについて基礎統計解析を行い最短距離制御の指標とした。また稲列の曲がりについて周波数特性を調べるためにスペクトル分析を行い、制御系の入力周波数を把握した。

第Ⅲ章では、各制御装置と角速度センサによる旋回角検出方法について述べた。供試コンバインの制御装置には左右の旋回を行う操向制御、前進、停止及び後進を行う変速制御、そして回行時に行う刈取り部の昇降制御の3系統とし、すべて油圧装置によって駆動するように供試コンバインを改造し、マイクロコンピュータによって制御した。

マイクロコンピュータ (Z80A) は2個使用し、それぞれ旋回角を演算するサブCPU部とコンバイン本体の制御を行うメインCPU部とから構成されており、メインCPU部はサブCPU部に対して旋回角データの転送要求を行い、得られたデータにより回行制御及び最短距離制御を行う。

角速度センサは米国Watson社製でピエゾエレクトリック効果を持つ結晶体でできた片持ちばりの振動を利用し、測定軸まわりに生じる角速度に比例したアナログ電圧の信号と、回転方向の情報を共に発生するセンサであり、その仕様は最大感度 $\pm 100^\circ/\text{S}/\text{VOLT}$ 、分解能 $0.1^\circ/\text{S}$ である。旋回角は角速度を時間で積分することによって求められるが、マイクロコンピュータで行う数値積分には誤差を伴う。この誤差は入力周波数とサンプリング周期の関数と見ることができ入力周波数が低いほど、またサンプリング周期が短いほど少なくなる。角速度センサの周波数応答 (70Hz) からすればサンプリング周期は5msとなり、8BitCPUでは演算不可能である。そこで積分誤差とサンプリング周期、入力周波数の関係を調べるためにサイン波を入力波形とし、この角速度について理論、数値積分を行い、積分誤差について検討した結果、演算時間の制約からサンプリング周期200msとした場合、入力周波数1Hz以下で最大積分誤差6%以下に抑えられることが明らかになった。稲列の周波数分析の結果よりコンバインへの入力周波数は非常に低いため、サンプリング周期200msで実用可能であ

ると言える。またサンプリング時間の違いによる誤差についても検討した。

プログラムはシステムの初期化後、サンプリング周期を発生するPITをスタートさせ200ms経過後、角速度センサの出力をA/D変換してCPUに取り込み、積分し200ms間の旋回角を求めてメモリに積算する。この数値を8桁の符号付BCDコードに変換して、中位4桁を取り出し0.1°単位の値としてメモリにストアしLEDディスプレイに表示する。最後に外部からのデータ転送要求を調べ、一巡のルーチンを終了する。これを200ms毎に繰り返し旋回角を演算する。

次に角速度センサを実機に搭載して、旋回角演算精度と振動による影響を調べた。実験はメインCPU部で角度と方向をソフトウェアで設定し、サブCPU部から転送されるデータとを比較することにより一定角度旋回させ、そのクローラの軌跡から実際に旋回した角度を測定し設定角度との差について検討した。

測定データを分散分析によって解析した結果、誤差は速度による影響が最も強く脱穀部駆動による影響は少ないことがわかった。これは速度の増加に伴う油圧系統の遅れと機体の慣性の影響が原因と考えられる。しかしながら、実際には速度は一定に設定されるため、この速度について角度設定を適切な値にすることで、誤差は小さくすることが可能である。

またこの実験はパーソナルコンピュータによるシミュレーションを行い同様の結果を確認した。コンバインの振動をシミュレートするために、停止状態ではあるが各部を運転し角速度センサの出力をスペクトラムアナライザで周波数分析した。コンバインの運動は基本的に座標変換による線形近似で行い走行面である土壌の影響は無視した。サンプリング周期を50msとした場合についてもシミュレーションを行い、より高精度になることを確認した。

次に第IV章では、旋回角検出方法を利用した回行制御について述べた。

ほ場末端部の検出は全ての操向センサによって稲の有無を監視し確実な検出を可能とした。ほ場末端部に達すると刈取り部を上昇し角速度センサによるフィードバックにより一定角度左旋回する。そしてディバイダの左外側のセンサ

が稲のコーナを検出するか設定時間が経過するまで後進し、一定角度右旋回し刈取り部を下降し停止する。このようにして 90° 旋回し次の稲列に進入するが、一定時間内に稲を検出しなければ刈取り終了と判断して停止する。このように回行制御はフィードバック制御されるが後進時間のみデータ設定を必要とするため土壌条件に応じてほ場にてキーボードから変更が可能とした。

実際のほ場において回行制御を行いコーナの形状を変化させた8つの実験区を設定し、その走行軌跡と稲の切株の位置を測定し、回行の成否、コーナ形状の影響等を検討した。回行動作を分割して各動作別に検討した結果、ほ場末端部を検出して旋回に必要な距離だけ前進するが、この距離と方向がコーナの形状に大きく影響されるため、回行終了後の位置にバラツキを生じることが明確となった。しかしながら、次の稲列進入時に操向センサによる補正によりコーナが鋭角で最初の進行方向が外側寄りの場合を除いてほぼ対応可能であることがわかった。

第V章では、最短距離制御の方法と実際の走行結果について述べた。

稲列のデータ収集はディバイダ先端の左右の内側に設置したセンサによって稲列に追従し、操舵間の時間を0.1秒単位でまた操舵角を角速度センサによって計測し、順次マイクロコンピュータのメモリにストアして稲列のデータとする。

最短距離演算はこのデータを直角座標系に変換し、条件に従い変曲点を抽出し、この点までの時間と距離を計算し、メモリにストアする。このようにして計算されたデータにより最短距離走行を行う。

実験は実際の稲の刈取りは行わず、コンバインを速度0.5m/sで手動操作により稲列データ収集を再現しその軌跡を20cm間隔で測定した。次に最短距離走行を行い、同様に軌跡を測定し、比較検討した。また最短距離演算を再現するために稲列収集データをパーソナルコンピュータに転送し、同一演算ルーチンで最短距離演算を行った。

稲列データによるメモリ上の走行軌跡は実際の走行軌跡をほぼ再現しており、最短距離演算も正確に計算されていることが確認された。実際の最短距離走行

は頂点における偏差の平均が約30cmであり、距離が長いほど偏差が小さくなる傾向があり、かなりよく再現されていると言える。今後は自動走行プログラムに本プログラムを組み込むことにより走行に関する一つのシステムが完成するが、実際の稲への適用には操向センサによる補正を加えることが必要である。また旋回角検出装置のサンプリング周期を20ms程度に短縮し、サンプリング時間による誤差をほとんど無視することが可能となれば飛躍的に精度の向上が期待でき、さらに非接触で安価な走行距離センサが開発されればコンバインの位置を完全に把握することが可能となり、移動体としての自立が可能となる。このため単に収穫作業の自動化に限らず、ボタン一つで車庫から出発し収穫を終了後、また車庫に帰着することも夢ではないと考えられる。

一方、テレビカメラによる画像認識技術の応用により、他のセンサを全く必要とせず、走行に関する自動化が可能となるが、膨大な情報をリアルタイムで処理する必要があり高速処理可能なマイクロコンピュータとソフトウェアの開発が最大のネックとなることが予想されるが、この分野における技術の進歩は目覚しく、将来的に最も有望であると考えられる。

謝 辞

本研究の遂行と取りまとめを終始、ご指導・ご教示を賜った、神戸大学農学部西村功教授、本論文の取りまとめに際しご教示を賜った、神戸大学工学部木村雄吉教授、同川井良次教授に衷心より感謝の意を表します。

また、実験の実施にあたりご協力をいただいた、神戸大学農学部川村恒夫助教授はじめ同圃場機械学研究室の各位に謝意を表します。

最後に供試コンバインを快く供与していただいたヤンマー農機㈱に感謝の意を表します。

参 考 文 献

- 1) 川村登・岡本嗣男：ロータリ耕うん部のトルク制御（第2報）、農機誌 30-2, 79 (1968)
- 2) Hesse, H., Möller, R.: Eine Elektrohydraulische Zwei-Größen-Tiefenregelung für grosse Schlepperanbaupflüge. Teil 1; Systeme zur Ein- und Zwei-Größen-Tiefenregelung, Grundl. Landtechnik, 22(3), 75-79 (1972)
- 3) Hesse, H., Möller, R.: Eine Elektrohydraulische Zwei-Größen-Tiefenregelung für grosse Schlepperanbaupflüge. Teil 2; Systemverhalten Versuchsergebnisse, Grundl. Landtechnik, 22(4), 102-106 (1972)
- 4) Dyck, F. B.: Automatic Depth Control for a Discer, Canadian Agric. Engng. 17(1), 47-49 (1975)
- 5) Hesse, H.: Signalverarbeitung in Pflugregelsystemen, Grundl. Landtechnik, 32, 54-59 (1982)
- 6) 川村登・藤浦建史：ロータリ耕うんトラクタの自動制御（第2報）、農機誌、41-3, 397-403 (1979)
- 7) 藤浦建史・川村登・児島弘明：ロータリ耕うんトラクタの自動制御—マイクロコンピュータによる走行速度・耕深制御—、農機学会第39回講演要旨, 14 (1980)
- 8) 伊藤信孝・竹内章博・馬淵久志・山本昭裕：トラクタのすべり率負荷制御に関する研究、農機学会関西支部報、48, 19-21 (1980)
- 9) 梅田重夫・穂波信雄：マイクロプロセッサによる油圧駆動トラクタのロータリ耕うん負荷制御—その1—、農機学会第40回講演要旨, 50 (1981)
- 10) Heitsu, D. C.: The Self-Propelled Hillside Combine, Agric. Engng., 37 182-183 (1956)
- 11) Pool, S. D.: Controls for Full-Leveling Hillside Combine, Agric. Engng., 37, 245-248 (1956)
- 12) 田尻功郎・伊佐務・野呂茂生・岩田芳幸：傾斜地用トラクタの試作研究（第16報）—加速度入力サーボ弁の固有特性について—、農機学会関西支部報、46, 21-23 (1979)
- 13) L. A. Liljedahl, J. Stait: Automatic Tractor Steering, Agric. Engng. 332-335 (1962)
- 14) M. A. Grovum, G. C. Zoerb: An Automatic Guidance System for Farm Tractor, Trans. ASAE, 565-573 (1970)

- 15) R. L. Parish, C. E. Coering: Developing an Automatic Steering system for a Hydraulic Vehicle, Trans. ASAE., 523-527 (1970)
- 16) A. P. Julian: Design, Performance of a Steering Control System for Agricultural Tractors, J. Agric. Engng. Res., 16, 332-336 (1971)
- 17) M. G. R. Warner, G. O. Harris: An Ultrasonic Guidance System for Driverless Tractor, J. Agric. Engng. Res., 17, 1-9 (1972)
- 18) M. B. Widden, J. R. Blair: A New Automatic Tractor Guidance System J. Agric. Engng. Res., 10-12 (1972)
- 19) K. Rushing: Developing the Driverless Tractor, Agric. Engng. 52-5, 260-262 (1971)
- 20) D. J. Hilton, A. A. W. Chestney: Low-cost Self-steering Devices for Out-of-Furrow Ploughing. Agric. Engng., 28-3, 102-106 (1973)
- 21) Induktives Fahrzeugleitsystem zur automatischen Fuehrung von Fahrzeugen ueber eine Flacehe (Patent-Auslegeschriften), Grundl. Landtechnik, 25-6, 194 (1975)
- 22) T. G. Kirk, A. E. Krause: Swather Edge Guide Steering Control System Trans. ASAE, 19-4, 639-642, 646 (1976)
- 23) 八木茂・武長孝・津賀幸之介・瀬山健治・梶山道雄：ケーブル誘導方式による農業車両の自動操縦に関する研究，ビークル・オートメーションシンポジウム講演論文集、47-48 (1978)
- 24) R. L. Schafer, R. E. Young: An Automatic Guidance System for Tractors Trans. ASAE, 22-1, 46, 49, 56 (1979)
- 25) R. E. Young, R. L. Schafer: Vehicle Guidance Simulation for Digital Automatic Control, Trans. ASAE, 22-4, 690-693 (1979)
- 26) 田中孝・田野信博・内藤孝：農用トラクタの自動操縦に関する研究、農機学会関西支部報、45, 1-4 (1979)
- 27) 田中孝・田野信博・内藤孝：農用トラクタの自動操縦に関する研究、農機学会関西支部報、46, 12-14 (1979)
- 28) 田中孝・田野信博・中嶋洋：オフ・ザ・ワイヤ方式による農用トラクタの自動操縦に関する研究（第1報）、農機学会関西支部報、49, 39-43 (1981)
- 29) G. O. Harries, B. Ambler: Automatic Ploughing: A Tractor Guidance System Using Opto-electronic Remote Sensing Techniques and Micro-processor Based Controller J. Agric. Engng. Res., 26-1, 33-53 (1981)

- 30) 穂波信雄・梅田重夫・平井宏昭：田植機の自動操向に関する研究、農機学会関西支部報、50,75-76 (1971)
- 31) 堀尾尚志・居垣千尋・佐々木圭一：クラブ・ステアリング車による自動耕うんの研究（第1報）、農機誌、43-4,523-531 (1982)
- 32) 堀尾尚志・居垣千尋・牧大助：クラブ・ステアリング車による自動耕うんの研究（第2報）、農機誌、44-3,423-429 (1983)
- 33) 堀尾尚志：On-off制御による逐次追従型自動操向の一方法、神戸大学農学部研究報告、16-1,245-251 (1984)
- 34) 堀尾尚志：ロータリ耕うんの自動操向におけるトラッキング・エラー値の検出、神戸大学農学部研究報告、16-2,477-482 (1985)
- 35) 池田善郎・川村登：コンバインの自動操向装置に関する研究、農機誌35-4,368-377 (1974)
- 36) 川村登・池田善郎・長沢信男：自脱型コンバインの自動操向に関する基礎的実験、農機学会関西支部報、30,40-41 (1971)
- 37) 伊藤信孝・杉本芳樹・小嶋潤一郎：コンバインのプログラム走行制御、農機学会関西支部報、40,46-48 (1976)
- 38) 伊藤信孝・伊谷一明・鬼頭孝治：自脱型コンバインの自動制御に関する研究－無線操縦制御の適用－、農機学会関西支部報、46,93-94 (1979)
- 39) Kaminski, T. L., Zoerb, G. C.: Automatic Header-height Control for Grain Crops, Trans. ASAE, 8-2, 284-287 (1965)
- 40) Bredfeldt, R. T.: Automatic Header Height Control for Self-Propelled Combines, Agric. Engng., 49, 666-667 (1968)
- 41) Story, A. G.: Remote Measurement of Concentration and Height of Heads of Standing Grain with Microwave Energy, Trans. ASAE, 13-1, 28-32 (1970)
- 42) Rehkugler, G. E.: Dynamic Analysis of Automatic Control of Combine Height, Trans. ASAE, 13-2, 225-231 (1970)
- 43) Schaller, R., et al.: Automatische Schnitthöhenregelung für den Mahdrescher E512, Agrart., 23, 368-370 (1973)
- 44) Pask, G. S., et al.: Automatic Header Height Control System for Windrowers, Trans. ASAE, 17, 597-602 (1974)
- 45) O. H. Friesen, et al.: For Combines; Controlling Feed Rates Automatically, J. Agric. Engng. Res., 47, 434-435 (1966)
- 46) 川村登・梅田幹男：油圧駆動コンバインの供給量制御（第2報）、農機誌、34-3, 236-241 (1973)

- 47) Eimer, M.: Funktion und Arbeitsqualität der drehzahlgeregelten Dreschtrommel, Grund. Landtechnik, 23(6), 158-163 (1973)
- 48) Graeber, E.: Wirtschaftliche und technische Aspekte der Kornverlust-
erfassung am Mähdrescher, Grundl. Landtechnik, 25(1), 15-17 (1975)
- 49) 川村恒夫・川村登・並河清：自脱型コンバインの適応制御（第1報）、
農機誌、38-2, 191-200 (1976)
- 50) 金藤裕治：無人化コンバイン—メカニズムと特性—、機械設計、20-5
117-121 (1974)
- 51) 鬼頭孝治・伊藤信孝：コンバインの自動走行制御、第4回ビークル・オ
ートメーション・シンポジウム講演論文集、29-32 (1981)
- 52) 鬼頭孝治・伊藤信孝：マイクロコンピュータ制御無人コンバインの開発
三重大学農学部学術報告、63, 235-255 (1981)
- 53) 喜多毅：マイクロコンピュータのコンバインへの応用：農業機械・施設
におけるマイクロコンピュータの利用技術講習会論文集、58-65 (1981)
- 54) Koji Kito, Isao Nishimura and Tsuneo Kawamura: FUNDAMENTAL STUDY
FOR OPTIMIZING AUTOMATIC STEERING CONTROL OF HEAD FEEDING TYPE
COMBINE, THE SCIENCE REPORTS OF FACULTY OF AGRICULTURAL, KOBE
UNIVERSITY, 16-1, 261-267 (1984)
- 55) 鬼頭孝治・西村功・川村恒夫：コンバインの最適制御（第2報）、第42
回農機学会講演要旨、53 (1984)
- 56) 日野幹雄：スペクトル解析、朝倉書店、210-223 (1981)
- 57) 川村恒夫：STUDIES ON ADAPTIVE CONTROL OF HEAD FEEDING TYPE COMBI-
NE, DOCTOR THESIS OF KYOTO UNIV., 6 (1980)
- 58) 日野幹雄：スペクトル解析、朝倉書店、31 (1981)
- 59) 五十嵐修：パネル制御の現状と動向、内燃機関、20-5, 91-99 (1981)
- 60) 田上勝利・高橋常夫・鈴木信彦・高橋文孝：自動車用慣性航法装置の開
発、計測と制御、21-7, 43-50 (1982)
- 61) 東口寛・保立和夫：ファイバジャイロの最近の傾向、第6回ビークル・
オートメーション・シンポジウム、1-9 (1983)
- 62) 古田勝久・中野道雄：デジタル制御システム（上巻）、ホルトサウン
ダース、267-269 (1984)
- 63) 片柳亮二：航空機の操縦システムにおけるデジタル制御、コンピュー
トロール NO. 2、コロナ社、97-102 (1983)
- 64) 西村功・川村恒夫・鬼頭孝治・荒木誠市：自脱コンバインの操向制御
農機学会関西支部報、56, 58-59 (1984)

- 65) 安居院猛・中嶋正之：デジタルシステム制御理論、産報出版、14-19 (1981)
- 66) 古田勝久・中野道雄：デジタル制御システム（上巻）、ホルトサウンダース、299 (1984)
- 67) 遠藤俊三・芝野保徳・千田芳宏・笹尾彰：自脱形コンバインの振動乗心地の実態、農機学会関西支部報、48,130-131 (1980)
- 68) W. Steeds: Tracked Vehicles, Automobile Engineer, 143 (1950)
- 69) M. G. Bekker: Theory of Land Locomotion, The Univ. of Michigan Press 363 (1956)
- 70) 林盤男：履帯スリップに基づく装軌車旋回理論実用的解析、日本機械学会講演論文集、740-12, 175-190 (1974)
- 71) 北野昌則・城崎博美：装軌車両の旋回性能に関する研究（第1報）、自動車技術会論文集、9, 51-57 (1975)
- 72) 北野昌則：装軌式トラクタの旋回理論の研究、農機誌、39-3, 271-278 (1978)
- 73) 江崎春雄 他3名：軟弱地盤におけるコンバインの走行性に関する研究、第42回農機学会講演要旨、2 (1983)
- 74) 坂井純 他5名：ゴムクローラ型コンバイン走行部の力学的特性に関する研究、第42回農機学会講演要旨、1 (1983)
- 75) 伊藤信孝：装軌車両の旋回時の運動に関する考察、三重大学農学部学術報告、67, 129-144 (1983)
- 76) 伊藤信孝・大塚雅司：上部旋回型コンバインの開発研究、第6回ビークル・オートメーション・シンポジウム講演論文集、51-54 (1982)
- 77) 伊藤信孝・大塚雅司：上部旋回式コンバインの開発研究、農機学会関西支部報、54, 96-98 (1983)
- 78) 江崎春雄他：新版農業機械ハンドブック、コロナ社、503 (1984)
- 79) 庄司渉・本田稔：Z80マイコンプログラムテクニック、電波新聞社、101-144 (1980)
- 80) 西沢昭：Z80上級プログラミング、CQ出版社、57-67 (1984)

付 録

メインCPU部及びサブCPU部
のアッセンブルリスト

旋回シミュレーション実験
のBASICプログラムリスト


```
*****
*
*   Turn Angle Calculating Program
*   for Sub Computer
*
*****
```

```

;
;
0000      START: EQU 0000H      ;PROGRAM START ADDRESS
1000      ADWK:  EQU 1000H      ;OFFSET DATA BUFFER
1003      IDATA: EQU 1003H      ;TURN ANGLE DATA BUFFER
1006      RDATA: EQU 1006H      ;COMPARE DATA BUFFER
0003      BYTE:  EQU 3         ;DATA BYTE NUMBER
1009      LACC:  EQU 1009H
100C      ACC:   EQU 100CH
100F      WRKM:  EQU 100FH
1012      WS:    EQU 1012H
1015      CARRY: EQU 1015H
1016      MINUS: EQU 1016H
1017      ZERO:  EQU 1017H
1018      OVRFLW: EQU 1018H
1019      DDATA: EQU 1019H      ;BCD DATA BUFFER
;
;
          ORG  START
0000 0600      LD  B,00H        ;DELAY FOR 8255
0002 00        NOP
0003 10FD      DJNZ $-1
0005 F3        INIT:  DI        ;INTERRUPT DISABLE
0006 310018    LD  SP,1800H     ;SP=1800H
0009 3E83      LD  A,83H        ;8255#0 MODE SET
000B D303      OUT (03H),A
000D 3E88      LD  A,88H        ;8255#1 MODE SET
000F D383      OUT (83H),A
0011 3E36      LD  A,36H        ;8253 MODE SET
0013 D387      OUT (87H),A
0015 3E01      LD  A,01H        ;8253 GATE=Hi
0017 D383      OUT (83H),A
0019 3E40      LD  A,40H        ;DIVIDED NUMBER=40000(DEC) 5Hz
001B D384      OUT (84H),A     ;LSB=40H
001D 3E9C      LD  A,9CH        ;MSB=9CH
001F D384      OUT (84H),A
0021 3EC0      LD  A,0C0H       ;BUSY, STROBE=HI
0023 D302      OUT (02H),A
0025 210000    LD  HL,0000H     ;IDATA AREA CLEAR
0028 220310    LD  (IDATA),HL
002B 221910    LD  (DDATA),HL   ;DDATA AREA CLEAR
002E 221B10    LD  (DDATA+2),HL
0031 AF        XOR  A
0032 320510    LD  (IDATA+2),A
;
;COUNT ROUTINE
;
0035 DB82      COUT: IN  A,(82H) ;200ms PAST?
0037 CB7F      BIT  7,A
0039 20FA      JR   NZ,COUT
003B DB82      C01:  IN  A,(82H)
003D CB7F      BIT  7,A
003F 28FA      JR   Z,C01
0041 CD4600    CALL ADCOV

```

```

0044 1818          JR    DIGF
;
0046 3E07      ADCOV: LD    A,07H          ;ADC CH=7
0048 D390          OUT   (90H),A
004A 0604          LD    B,04H          ;20 micro sec wait
004C 00          AD1:  NOP
004D 10FD          DJNZ  AD1
004F D391          OUT   (91H),A          ;START PULSE ON
0051 DB92      AD2:  IN    A,(92H)        ;END OF CONVERSION?
0053 E680          AND    80H
0055 20FA          JR    NZ,AD2
0057 DB92          IN    A,(92H)        ;MSB=H
0059 67          LD    H,A
005A DB93          IN    A,(93H)        ;LSB=L
005C 6F          LD    L,A
005D C9          RET
;
;ADC DATA TRANSFORM ROUTINE
;INPUT PARAMETER DE=ADC DATA
;
005E E5          DIGF: PUSH  HL          ;DIGITAL FILTER
005F 7C          LD    A,H
0060 C85F          BIT    3,A
0062 200B          JR    NZ,C1
0064 11F807      LD    DE,7F8H          ;HL<7F8H
0067 AF          XOR    A
0068 ED52          SBC   HL,DE
006A F27900      JP    P,C2
006D 180E          JR    C3
006F 110708      C1:  LD    DE,807H          ;HL>807H
0072 AF          XOR    A
0073 ED52          SBC   HL,DE
0075 3802          JR    C,C2
0077 1804          JR    C3
0079 E1          C2:  POP   HL
007A C35001      JP    HSR
;
007D E1          C3:  POP   HL
007E 110008      CALC: LD    DE,800H          ; (AD DATA - 800H)
0081 AF          XOR    A          ; 000-FFF  → F800-07FF  |
0082 ED52          SBC   HL,DE          ;          |
0084 7C          LD    A,H          ; FFF800-0007FF ← |
0085 CB7F          BIT    7,A
0087 2006          JR    NZ,CA1
0089 AF          XOR    A
008A 320210      LD    (ADWK+2),A
008D 1805          JR    CA2
008F 3EFF      CA1:  LD    A,0FFH
0091 320210      LD    (ADWK+2),A
0094 220010      CA2:  LD    (ADWK),HL
;
;HEX 677 / カサ ROUTINE
; ROUTINE NAME:LADD
;
0097 110010      LADD: LD    DE,ADWK
009A 210310      LD    HL,IDATA
009D 0603          LD    B,BYTE
009F AF          XOR    A
00A0 1A          LA1:  LD    A,(DE)
00A1 8E          ADC    A,(HL)

```

```

00A2 77          LD    (HL),A
00A3 13          INC  DE
00A4 23          INC  HL
00A5 10F9       DJNZ LA1

;
;DATA COMPARE ROUTINE
;
00A7 3A0510     COMPR: LD  A,(IDATA+2)
00AA CB7F       BIT  7,A
00AC 2035       JR   NZ,CMP2
00AE 110310     CSUB1: LD  DE,IDATA
00B1 210801     LD  HL,PDATA
00B4 DD210610   LD  IX,RDATA
00B8 0603       LD  B,BYTE
00BA AF         XOR  A
00BB 1A         CSUB2: LD  A,(DE)
00BC 9E         SBC  A,(HL)
00BD DD7700     LD  (IX),A
00C0 13         INC  DE
00C1 23         INC  HL
00C2 DD23       INC  IX
00C4 10F5       DJNZ CSUB2
00C6 FA1601     JP   M,CSUB3
00C9 110610     LD  DE,RDATA
00CC 210801     LD  HL,NDATA
00CF DD210310   LD  IX,IDATA
00D3 0603       LD  B,BYTE
00D5 AF         XOR  A
00D6 1A         CSUB4: LD  A,(DE)
00D7 8E         ADC  A,(HL)
00D8 DD7700     LD  (IX),A
00DB 13         INC  DE
00DC 23         INC  HL
00DD DD23       INC  IX
00DF 10F5       DJNZ CSUB4
00E1 1833       JR   CSUB3
00E3 110801     CMP2: LD  DE,NDATA
00E6 210310     LD  HL,IDATA
00E9 DD210610   LD  IX,RDATA
00ED 0603       LD  B,BYTE
00EF AF         XOR  A
00F0 1A         CSUB5: LD  A,(DE)
00F1 9E         SBC  A,(HL)
00F2 DD7700     LD  (IX),A
00F5 13         INC  DE
00F6 23         INC  HL
00F7 DD23       INC  IX
00F9 10F5       DJNZ CSUB5
00FB FA1601     JP   M,CSUB3
00FE 110801     LD  DE,PDATA
0101 210610     LD  HL,RDATA
0104 DD210310   LD  IX,IDATA
0108 0603       LD  B,BYTE
010A AF         XOR  A
010B 1A         CSUB7: LD  A,(DE)
010C 9E         SBC  A,(HL)
010D DD7700     LD  (IX),A
0110 13         INC  DE
0111 23         INC  HL
0112 DD23       INC  IX

```

```

0114 10F5          DJNZ CSUB7
0116 210310       CSUB3: LD  HL, IDATA
0119 110C10       LD  DE, ACC
011C 010300       LD  BC, BYTE
011F ED80         LDIR
0121 111910       LD  DE, DDATA
0124 CDE201       CALL B2BCDX
0127 AF           XOR  A           ;DDATA*10
0128 211910       LD  HL, DDATA
012B ED6F         RLD
012D 23          INC  HL
012E ED6F         RLD
0130 23          INC  HL
0131 ED6F         RLD
0133 3A1C10       DISP: LD  A, (DDATA+3)   ;SIGN=1 OR 0
0136 CB7F         BIT  7, A
0138 2806         JR   Z, DIS1
013A 3E03         LD  A, 03H           ;DISPLAY '-'
013C D383         OUT (83H), A
013E 1804         JR   DIS2
0140 3E02         DIS1: LD  A, 02H           ;NO DISPLAY '-'
0142 D383         OUT (83H), A
0144 3A1B10       DIS2: LD  A, (DDATA+2)   ;100,10 DISPLAY
0147 CB8F         RES  7, A
0149 D381         OUT (81H), A
014B 3A1A10       LD  A, (DDATA+1)   ;1,.1 DISPLAY
014E D380         OUT (80H), A
                ;
                ;DATA HANDSHAKE ROUTINE
                ;
0150 D802         HSR:  IN  A, (02H)
0152 CB47         BIT  0, A           ;DRQ1?  TURN ANGLE REQUEST
0154 280B         JR   Z, HSR1
0156 CB4F         BIT  1, A           ;DRQ2?  A/D DATA REQUEST
0158 2850         JR   Z, HSR3
015A CB57         BIT  2, A           ;DRQ3?  SYSTEM RESET REQUEST
015C 2869         JR   Z, HSR6
015E C33500       JP   COUT
0161 3A1C10       HSR1: LD  A, (DDATA+3)
0164 CB7F         BIT  7, A
0166 2808         JR   Z, HSR9
0168 3A1B10       LD  A, (DDATA+2)
0168 F680         OR   80H
016D 321B10       LD  (DDATA+2), A
0170 211A10       HSR9: LD  HL, DDATA+1   ;SEND DATA ADDRESS SET
0173 00          NOP
0174 DB02         HSR2: IN  A, (02H)   ;DRQ=LOW?
0176 CB47         BIT  0, A
0178 20FA         JR   NZ, HSR2
017A 3E0E         LD  A, 0EH           ;BUSY=LOW
017C D303         OUT (03H), A
017E 7E          LD  A, (HL)           ;LSB DATA SET
017F D300         OUT (00H), A
0181 3E0C         LD  A, 0CH           ;STROBE=LOW
0183 D303         OUT (03H), A
0185 DB02         HSR8: IN  A, (02H)   ;DRQ=HI?
0187 CB47         BIT  0, A
0189 28FA         JR   Z, HSR8
018B 3E0D         LD  A, 0DH           ;STROBE=HI
018D D303         OUT (03H), A

```

```

018F 23          INC  HL
0190 7E          LD   A,(HL)          ;MSB DATA SET
0191 0300        OUT  (00H),A
0193 00          NOP
0194 3E0C        LD   A,0CH          ;STROBE=LOW
0196 D303        OUT  (03H),A
0198 0B02        HSR10: IN  A,(02H)      ;DRQ=LOW?
019A CB47        BIT  0,A
019C 20FA        JR   NZ,HSR10
019E 00          NOP
019F 3E0D        LD   A,0DH          ;STROBE=HI
01A1 D303        OUT  (03H),A
01A3 3E0F        LD   A,0FH          ;BUSY=HI
01A5 D303        OUT  (03H),A
01A7 C33500      JP   COUT
01AA CD4600      HSR3:  CALL ADCOV
01AD 3E02        LD   A,02H          ;NO DISPLAY '-'
01AF D383        OUT  (83H),A
01B1 7C          LD   A,H          ;AD DATA DISPLAY
01B2 D381        OUT  (81H),A
01B4 7D          LD   A,L
01B5 D380        OUT  (80H),A
01B7 1600        LD   D,00H        ;TIMER DELAY
01B9 0600      HSR4:  LD   B,00H
01BB 00          NOP
01BC 10FD        DJNZ $-1
01BE 15          DEC  D
01BF 20F8        JR   NZ,HSR4
01C1 0B02        IN  A,(02H)
01C3 CB57        BIT  2,A
01C5 20E3        JR   NZ,HSR3
01C7 3E0E        HSR6:  LD   A,0EH          ;BUSY=LOW
01C9 D303        OUT  (03H),A
01CB 0B02        HSR7:  IN  A,(02H)      ;DRQ2=HI?
01CD CB57        BIT  2,A
01CF 28FA        JR   Z,HSR7
01D1 3E0F        LD   A,0FH          ;BUSY=HI
01D3 D303        OUT  (03H),A
01D5 C30500      JP   INIT

01D8 504600      ; PDATA: DB 50H,46H,00H          ;+180.00
01DB 0B09FF      ; NDATA: DB 0B0H,0B9H,0FFH        ;-180.00
01DE 64000000    ; B100: DW 100,0

;
; BINARY TO BCD FORMAT ( 3 BYTE BINARY -----> 4 BYTE BCD )
; SUBROUTINE NAME B2BCDX
;
01E2 CD3302      B2BCDX:CALL BINIT
01E5 060A        LD   B,10
01E7 210F10      LD   HL,WRKM
01EA 3600        B2DX1: LD  (HL),0
01EC 23          INC  HL
01ED 10FB        DJNZ B2DX1
01EF 3A0E10      LD   A,(ACC+BYTE-1)
01F2 CB7F        BIT  7,A
01F4 0E00        LD   C,0
01F6 280A        JR   Z,B2DX2
01F8 D5          PUSH DE
01F9 110C10      LD   DE,ACC
01FC CD4302      CALL NEGAT

```

```

01FF D1          POP  DE
0200 0EFF          LD   C,0FFH
0202 0604          B2DX2: LD   B,BYTE+1
0204 05            B2DX3: PUSH DE
0205 110E01         LD   DE,B100
0208 CD5202         CALL DIV
0208 210910         LD   HL,LACC
020E 01            POP  DE
020F CDA602         CALL B2BCD
0212 13            INC  DE
0213 10EF          DJNZ B2DX3
0215 18            DEC  DE
0216 3A1710         LD   A,(ZERO)
0219 87            OR   A
021A 2805          JR   Z,B2DX4
021C 1A            LD   A,(DE)
021D E6F0          AND  0F0H
021F 2802          JR   Z,B2DX5
0221 3EFF          B2DX4: LD   A,0FFH
0223 321810        B2DX5: LD   (OVRFLW),A
0226 1A            LD   A,(DE)
0227 E67F          AND  7FH
0229 CB79          BIT  7,C
022B 2802          JR   Z,B2DX6
022D CBFF          SET  7,A
022F 12            B2DX6: LD   (DE),A
0230 C33E02        JP   EXIT

;
;SAVE REGISTER
;
0233 E3            BINIT: EX  (SP),HL
0234 05            PUSH DE
0235 C5            PUSH BC
0236 F5            PUSH AF
0237 E5            PUSH HL
0238 210C10        LD   HL,ACC
0238 0603          LD   B,BYTE
0230 C9            RET

;
;POPUP REGISTER
;
023E F1            EXIT: POP  AF
023F C1            POP  BC
0240 D1            POP  DE
0241 E1            POP  HL
0242 C9            RET

;
;NEGATE (DE)---(DE)*(-1)
;
0243 CD3302        NEGAT: CALL BINIT
0246 37            SCF
0247 1A            NEGATO:LD  A,(DE)
0248 2F            CPL
0249 CE00          ADC  A,0
024B 12            LD   (DE),A
024C 13            INC  DE
024D 10F8          DJNZ NEGATO
024F C33E02        JP   EXIT

;
;DIVIDE ACC BY (DE)

```

```

;
0252 CD3302 DIV: CALL BINIT
0255 CDCE02 CALL POSNEG
0258 AF XOR A
0259 210F10 LD HL,WRKM
025C 0603 LD B,BYTE
025E B6 OR (HL)
025F 23 INC HL
0260 10FC DJNZ $-2
0262 2008 JR NZ,DIVO
0264 3EFF LD A,OFFH
0266 321810 LD (OVRFLW),A
0269 C33E02 JP EXIT
026C 210C10 DIV0: LD HL,ACC
026F 110F10 LD DE,WRKM
0272 0618 LD B,24 ;24 Bit
0274 CDF402 DIV1: CALL SFTLD
0277 CD5D03 CALL COMP
027A 3A1610 LD A,(MINUS)
027D A7 AND A
027E 200B JR NZ,DIV2
0280 CD0E03 CALL SUB
0283 3A0910 LD A,(LACC)
0286 CBC7 SET 0,A
0288 320910 LD (LACC),A
028B 10E7 DIV2: DJNZ DIV1
028D CD4603 CALL XLAUA
0290 CB41 BIT 0,C
0292 110C10 LD DE,ACC
0295 C44302 CALL NZ,NEGAT
0298 CB79 BIT 7,C
029A 110910 LD DE,LACC
029D C44302 CALL NZ,NEGAT
02A0 CD2803 CALL TEST
02A3 C33E02 JP EXIT
;
;BINARY TO BCD (1BYTE)
;
02A6 F5 B2BCD: PUSH AF
02A7 C5 PUSH BC
02A8 0600 LD B,0
02AA 7E LD A,(HL)
02AB D664 B2D1: SUB 100
02AD 04 INC B
02AE 30FB JR NC,B2D1
02B0 05 DEC B
02B1 C664 ADD A,100
02B3 0E00 LD C,0
02B5 D60A B2D2: SUB 10
02B7 0C INC C
02B8 30FB JR NC,B2D2
02BA 0D DEC C
02BB C60A ADD A,10
02BD CB01 RLC C
02BF CB01 RLC C
02C1 CB01 RLC C
02C3 CB01 RLC C
02C5 B1 OR C
02C6 12 LD (DE),A
02C7 78 LD A,B

```

```

02C8 13          INC  DE
02C9 12          LD   (DE),A
02CA 18          DEC  DE
02CB C1          POP  BC
02CC F1          POP  AF
02CD C9          RET

;
;PREPARTION & CHANGE TO POSITIVE
;
02CE 0E00        POSNEG:LD  C,0
02D0 CD7F03      CALL  WMLACC
02D3 3A0810      LD   A,(LACC+BYTE-1)
02D6 CB7F        BIT   7,A
02D8 110910      LD   DE,LACC
02DB 2806        JR   Z,PN1
02DD CD4302      CALL  NEGAT
02E0 0C          INC  C
02E1 CBF9        SET  7,C
02E3 3A1110      PN1:  LD  A,(WRKM+BYTE-1)
02E6 CB7F        BIT   7,A
02E8 110F10      LD   DE,WRKM
02EB 2806        JR   Z,PN2
02ED CD4302      CALL  NEGAT
02F0 0C          INC  C
02F1 CBF1        SET  6,C
02F3 C9          PN2:  RET

;
;SHIFT LEFT ACC & LACC
;
02F4 CD3302      SFTLD: CALL  BINIT
02F7 0606        LD   B,BYTE+BYTE
02F9 210910      LD   HL,LACC
02FC AF          XOR  A
02FD 7E          SFTL1: LD  A,(HL)
02FE 17          RLA
02FF 77          LD   (HL),A
0300 23          INC  HL
0301 10FA        DJNZ SFTL1
0303 3E00        LD   A,0
0305 3001        JR   NC,SFTR3
0307 2F          CPL
0308 321510      SFTR3: LD  (CARRY),A
030B C33E02      JP   EXIT

;
;SUB (DE) FROM ACC
;
030E CD3302      SUB:  CALL  BINIT
0311 AF          SUB0:  XOR  A
0312 EB          EX   DE,HL
0313 1A          SUB1:  LD   A,(DE)
0314 9E          SBC  A,(HL)
0315 12          LD   (DE),A
0316 23          INC  HL
0317 13          INC  DE
0318 10F9        DJNZ SUB1
031A 3E00        ADD2:  LD  A,0
031C 3001        JR   NC,ADD3
031E 2F          CPL
031F 321510      ADD3:  LD  (CARRY),A
0322 CD2803      CALL  TEST

```



```

0325 C33E02          JP   EXIT
                    ;
                    ;TEST ACC AND SET FLAG
                    ;
0328 210B10        TEST: LD   HL,ACC-1
032B 0603          LD   B,3
032D AF           XOR   A
032E 23           TEST1: INC  HL
032F B6           OR   (HL)
0330 10FC         DJNZ TEST1
0332 3E00         LD   A,0
0334 2001         JR   NZ,TEST2
0336 2F           CPL
0337 321710       TEST2: LD   (ZERO),A
033A 7E           LD   A,(HL)
033B A7           AND  A
033C 3E00         LD   A,0
033E F24203       JP   P,TEST3
0341 2F           CPL
0342 321610       TEST3: LD   (MINUS),A
0345 C9           RET
                    ;
                    ;EXCHANGE ACC LACC
                    ;
0346 CD3302       XLAUA: CALL BINIT
0349 210C10       LD   HL,ACC
034C 110910       LD   DE,LACC
034F 0603         LD   B,BYTE
0351 4E           XLAUA1:LD   C,(HL)
0352 1A          LD   A,(DE)
0353 77          LD   (HL),A
0354 79          LD   A,C
0355 12          LD   (DE),A
0356 23          INC  HL
0357 13          INC  DE
0358 10F7        DJNZ XLAUA1
035A C33E02       JP   EXIT
                    ;
                    ;COMPARE ACC TO WRKM
                    ;
035D CD3302       COMP:  CALL BINIT
0360 010300       LD   BC,BYTE
0363 210C10       LD   HL,ACC
0366 111210       LD   DE,WS
0369 EDB0         LDIR
036B 110F10       LD   DE,WRKM
036E CD0E03       CALL SUB
0371 110C10       LD   DE,ACC
0374 211210       LD   HL,WS
0377 010300       LD   BC,BYTE
037A EDB0         LDIR
037C C33E02       JP   EXIT
                    ;
                    ;MOVE (DE) TO MORKING MEMORY
                    ;MOVE ACC TO LACC AND CLEAR ACC
                    ;
037F C5           WMLACC:PUSH BC
0380 E5           PUSH HL
0381 EB           EX   DE,HL
0382 110F10       LD   DE,WRKM

```

```
0385 010300      LD   BC,BYTE
0388 EDB0        LDIR
038A 210C10      LD   HL,ACC
038D 110910      LD   DE,LACC
0390 010300      LD   BC,BYTE
0393 EDB0        LDIR
0395 0603        LD   B,BYTE
0397 AF          XOR  A
0398 12          WMLAC1:LD  (DE),A
0399 13          INC  DE
039A 10FC        DJNZ WMLAC1
039C E1          POP  HL
039D C1          POP  BC
039E C9          RET
```

```

*****
*
*   Combine Control Program   *
*     Main computer         *
*
*****

```

```

;
;
0000      START: EQU 0000H      ;START ADDRESS
0038      MAKINT: EQU 0038H     ;INT ENTRY ADDRESS
0066      TIMINT: EQU 0066H     ;NMI ENTRY ADDRESS
1000      DATNUM: EQU 1000H     ;DATA PAIR NUMBER
1001      DATCNT: EQU 1001H     ;DATA COUNTER TOP ADDRESS
1002      DATBUF: EQU 1002H     ;DATA BUFFER FOR TIME & TURN ANGLE
1200      XYIBUF: EQU 1200H    ;X-Y DATA BUFFER
1400      SAIADR: EQU 1400H    ;CHOOSD X-Y DATA BUFFER
1600      ATOBJD: EQU 1600H    ;OBJECTIVE ANGLE & TIME BUFFER
3800      STKPOI: EQU 3800H    ;STACK POINTER
3000      TIMCNT: EQU 3000H    ;TIMER COUNTER 0.1sec
3002      TIMBUF: EQU 3002H    ;TIMER BUFFER
3004      DACTAD: EQU 3004H    ;DATA COUNTER ADDRESS BUFFER
3006      DATADR: EQU 3006H    ;NEXT DATA ADDRESS BUFFER
3008      BATMDA: EQU 3008H    ;BACK TIME DATA ADDRESS
300A      TBUFF: EQU 300AH    ;POSITION BUFFER
300C      SDATA: EQU 300CH    ;TURN ANGLE DATA ADDRESS RECIEVED
300E      BCDDAT: EQU 300EH    ;DATA BUFFER FOR ZDB2B
3010      ANG1: EQU 3010H     ;BINARY TURN ANGLE DATA FOR SAITAN
3012      TIMS: EQU 3012H
3014      TIMB: EQU 3014H     ;BINARY TIME DATA FOR SAITAN
3016      ANGS: EQU 3016H     ;TRAVELING DIRECTIONAL ANGLE
3018      XID: EQU 3018H      ;X-DATA AERA
301A      YID: EQU 301AH      ;Y-DATA AERA
301C      ANGLE: EQU 301CH    ;TURN ANGLE BUFFER FOR DATA ACQ.
301E      CPDAT1: EQU 301EH
3020      CPDAT2: EQU 3020H
3022      DEFDA1: EQU 3022H
3024      DEFDA2: EQU 3024H
3026      SAIDAT: EQU 3026H
3028      YNDAT: EQU 3028H
302A      YOBA1: EQU 302AH
302C      YOBA2: EQU 302CH
302E      DEFYOB: EQU 302EH
3030      DEFXOB: EQU 3030H
3032      OBTHI1: EQU 3032H
3024      OBTHI2: EQU 3024H
3036      XSQUR: EQU 3036H
3038      YSQUR: EQU 3038H
303A      SQUR: EQU 303AH
303C      TRFLG: EQU 303CH
;
;
;
;

```

```

ORG START

```

```

0000 ED56      INIT: IM 1      ;INTERRUPT MODE 1
0002 0600      LD B,00H      ;DELAY FOR 8255 #1
0004 00        NOP
0005 10FD      DJNZ $-1
0007 310038    LD SP,STKPOI  ;SET STACK POINTER
000A 3E9A      LD A,9AH     ;8255 #0 MODE SET

```

```

000C D303          OUT  (03H),A
000E 3E92          LD   A,92H          ;8255 #1 MODE SET
0010 D383          OUT  (83H),A
0012 3E07          LD   A,07H          ;8255 #0 DRQ1-3=HI
0014 D302          OUT  (02H),A          ;8651B R.TIME C. DISABLE
0016 3E40          LD   A,40H          ;SET NEUTRAL POSITION DATA
0018 320A30        LD   (TBUFF),A
001B 3E00          LD   A,0
001D 323C30        LD   (TRFLG),A
0020 FB           EI                   ;ENABLE INTERRUPT
0021 C3FE00        JP   INIT1

;
;KEY INPUT SUBROUTINE (WAIT UNTIL KEY IN)
;
00FE             STAKEY:EQU 0FEH          ;START KEY DATA
00FD             STPKEY:EQU 0FDH          ;STOP KEY DATA
00FB             BRKKEY:EQU 0FBH          ;BREAK KEY DATA
00F7             PG1KEY:EQU 0F7H          ;PROGRAM 1 KEY DATA
00EF             PG2KEY:EQU 0EFH          ;      2
00DF             PG3KEY:EQU 0DFH          ;      3
00BF             PG4KEY:EQU 0BFH          ;      4
0077             PG5KEY:EQU 77H          ;      5
006F             PG6KEY:EQU 6FH          ;      6
005F             PG7KEY:EQU 5FH          ;      7
003F             PG8KEY:EQU 3FH          ;      8
007F             SFTKEY:EQU 7FH          ;SHIFT KEY

;
0024 DB01         KEYINP: IN  A,(01H)          ;INPUT Acc TO KEY IN BUFFER
0026 FEFF         CP   0FFH          ;NOT KEY IN?
0028 28FA         JR   Z,KEYINP
002A FE7F         CP   7FH          ;SHIFT KEY?
002C 28F6         JR   Z,KEYINP
002E C9          RET

;
002F 3E0E         MINT4: LD   A,0EH          ;CONTINUE FROM MSKINT
0031 D383         OUT  (83H),A          ;BRK or TPF LAMP OFF
0033 3E0C         LD   A,0CH
0035 D383         OUT  (83H),A
0037 1828         JR   MINT5

;
;INTERRUPT ROUTINE
;
;          ORG  MAKINT
;
0038 D9          EXX          ;ALL Reg. PUSH
0039 08          EX   AF,AF'
003A 3A0A30      LD   A,(TBUFF)          ;PWS DATA SAVE
003D 57          LD   D,A
003E AF         XOR   A          ;ALL MOTION IS STOP
003F D382         OUT  (82H),A
0041 DB01         IN   A,(01H)          ;BRK or TPF?
0043 CB57         BIT   2,A
0045 2004         JR   NZ,MINT1
0047 3E0F         LD   A,0FH          ;BRK LAMP ON
0049 1802         JR   MINT2
004B 3E0D         MINT1: LD   A,0DH          ;TPF LAMP ON
004D D383         MINT2: OUT  (83H),A
004F 0E40         LD   C,NEUT          ;COMBINE STOP
0051 CD8600      CALL PWSHFT
0054 CD2400      MINT3: CALL KEYINP

```

```

0057 FEFE          CP   STAKEY
0059 20F9          JR   NZ,MINT3
005B 4A            LD   C,D
005C CD8600        CALL PWSHFT
005F 18CE          JR   MINT4
0061 08            MINT5: EX AF,AF'
0062 D9            EXX
0063 FB            EI
0064 C9            RET

;
;REAL TIME CLOCK COUNT ROUTINE (UP TO 999.9 SEC)
;
      ORG   TIMINT
;
0066 F5            PUSH AF                ;STORE ALL Reg.
0067 C5            PUSH BC
0068 D5            PUSH DE
0069 E5            PUSH HL
006A AF            XOR  A                ;CLEAR CARRY
006B 3A0030        LD   A,(TIMCNT)          ;TIME COUNTER LSB + 1
006E 3C            INC  A
006F 27            DAA                    ;DECIMAL ADJUST
0070 320030        LD   (TIMCNT),A          ;RESTORE
0073 3802          JR   C, TMINT1          ;CARRY ?
0075 1809          JR   TMINT2
0077 AF            TMINT1: XOR A          ;CLEAR CARRY
0078 3A0130        LD   A,(TIMCNT+1)        ;TIME COUNTER MSB + 1
007B 3C            INC  A
007C 27            DAA                    ;DECIMAL ADJUST
007D 320130        LD   (TIMCNT+1),A
0080 E1            TMINT2: POP HL          ;RESTORE ALL Reg.
0081 D1            POP  DE
0082 C1            POP  BC
0083 F1            POP  AF
0084 ED45          RETN

;
;POWER SHIFT CONTROL SUBROUTINE
;      INPUT PARAMETER  C reg.=POSITION
;
0008              THIRD: EQU 08H          ;3 SPEED
0010              SECOND: EQU 10H         ;2 SPEED
0020              FIRST: EQU 20H         ;1 SPEED
0040              NEUT: EQU 40H          ;NEUTRAL
0080              REVR5: EQU 80H         ;REVERSE
;
0086 D881          PWSHFT: IN  A,(81H)     ;INPUT POWER SHIFT POSITION
0088 E6F8          AND  0F8H
008A 2003          JR   NZ,PWS1          ;NO CONTACT ?
008C 3A0A30        LD   A,(TBUFF)        ;A=TIME BUFFER DATA
008F B9            PWS1: CP  C            ;SAME ?
0090 C8            RET  Z
0091 3006          JR   NC,PWS2
0093 3E0B          LD   A,DECON          ;DECELERATING
0095 D383          OUT  (83H),A
0097 1804          JR   PWS3
0099 3E09          PWS2: LD  A,ACCON      ;ACCELERATING
009B D383          OUT  (83H),A
009D D881          PWS3: IN  A,(81H)
009F E6F8          AND  0F8H
00A1 B9            CP  C                ;OBJECTIVE POSITION ?

```

```

00A2 20F9      JR    NZ,PWS3
00A4 320A30    PWS4: LD    (TBUFF),A
00A7 3E08      LD    A,ACCOF
00A9 D383      OUT   (83H),A
00AB 3E0A      LD    A,DECOF
00AD D383      OUT   (83H),A
00AF C9        RET

;
;SUBCPU HAND SHAKE SUBROUTINE
;
;
00B0 E5        HSSR:  PUSH HL
00B1 C5        PUSH BC
00B2 210C30    LD    HL,SDATA
00B5 AF        XOR   A                                ;REQUEST TURN ANGLE DATA
00B6 D303      OUT   (03H),A
00B8 DB02      HSSR2: IN   A,(02H)                        ;BUSY=LOW?
00BA CB7F      BIT   7,A
00BC 20FA      JR    NZ,HSSR2
00BE DB02      HSSR3: IN   A,(02H)                        ;STROBE=LOW?
00C0 CB77      BIT   6,A
00C2 20FA      JR    NZ,HSSR3
00C4 DB00      IN   A,(00H)                                ;READ LSB DATA
00C6 77        LD    (HL),A
00C7 3E01      LD    A,01H                                ;DRQ=HI
00C9 D303      OUT   (03H),A
00CB 0605      LD    B,05H                                ;DELAY
00CD 10FE      DJNZ $
00CF DB02      HSSR4: IN   A,(02H)                        ;STROBE=LOW?
00D1 CB77      BIT   6,A
00D3 20FA      JR    NZ,HSSR4
00D5 23        INC  HL
00D6 DB00      IN   A,(00H)                                ;READ MSB DATA
00D8 77        LD    (HL),A
00D9 AF        XOR   A                                ;DRQ=LOW
00DA D303      OUT   (03H),A
00DC DB02      HSSR5: IN   A,(02H)                        ;STROBE=HI?
00DE CB77      BIT   6,A
00E0 28FA      JR    Z,HSSR5
00E2 3E01      LD    A,01H                                ;DRQ=HI
00E4 D303      OUT   (03H),A
00E6 C1        POP  BC
00E7 E1        POP  HL
00E8 C9        RET

;
;SUBCPU RESET SUBROUTINE
;
;
00E9 3E04      HSRQ3: LD    A,04H                                ;DRQ3=LOW
00EB D303      OUT   (03H),A
00ED DB02      HDR1:  IN   A,(02H)                        ;BUSY=LOW?
00EF CB7F      BIT   7,A
00F1 20FA      JR    NZ,HDR1
00F3 3E05      LD    A,05H                                ;DRQ3=HI
00F5 D303      OUT   (03H),A
00F7 C5        PUSH BC
00F8 0610      LD    B,10H                                ;DELAY
00FA 10FE      DJNZ $
00FC C1        POP  BC
00FD C9        RET
;

```

```

;
;
*****
*                               *
* MAIN ROUTINE                 *
*                               *
*****
;
;
;SET INITIAL DATA FOR DATA ACQUISITION
;
00FE 210110  INIT1: LD   HL,DATCNT           ;SET DATA COUNTER TOP ADDRESS
0101 220430          LD   (DACTAD),HL
0104 AF          XOR   A                   ;CLEAR DATA PAIR NUMBER
0105 320010          LD   (DATNUM),A
0108 210210          LD   HL,DATBUF       ;SET DATA AREA TOP ADDRESS
010B 220630          LD   (DATADR),HL
010E 3E40          LD   A,40H           ;BACK TIME DATA
0110 320830          LD   (BATMDA),A
;
0113 3E0F          MAIN: LD   A,0FH       ;TPF AND BRK LAMP ON
0115 D383          OUT  (83H),A
0117 3E0D          LD   A,0DH
0119 D383          OUT  (83H),A
011B CD2400        MAIN1: CALL KEYINP
011E FEF7          CP   PG1KEY
0120 281E          JR   Z,MAIN2
0122 FEEF          CP   PG2KEY
0124 281A          JR   Z,MAIN2
0126 FEDF          CP   PG3KEY
0128 2816          JR   Z,MAIN2
012A FEBF          CP   PG4KEY
012C 2812          JR   Z,MAIN2
012E FE77          CP   PG5KEY
0130 280E          JR   Z,MAIN2
0132 FE6F          CP   PG6KEY
0134 280A          JR   Z,MAIN2
0136 FE5F          CP   PG7KEY
0138 2806          JR   Z,MAIN2
013A FE3F          CP   PG8KEY
013C 2802          JR   Z,MAIN2
013E 180B          JR   MAIN1
0140 F5           MAIN2: PUSH AF
0141 3E0E          LD   A,0EH           ;BRK LAMP OFF
0143 D383          OUT  (83H),A
0145 CD2400        MAIN3: CALL KEYINP
0148 FEFD          CP   STPKEY
014A 28C7          JR   Z,MAIN
014C FEFE          CP   STAKEY
014E 20F5          JR   NZ,MAIN3
0150 3E0C          LD   A,0CH           ;TPF LAMP OFF
0152 D383          OUT  (83H),A
0154 F1           POP   AF
0155 FEF7          CP   PG1KEY
0157 CA8004        JP   Z,AUTO1
015A FEEF          CP   PG2KEY
015C CA8001        JP   Z,AUTO2
015F FEDF          CP   PG3KEY
0161 CA3805        JP   Z,BKTIME
0164 FEBF          CP   PG4KEY

```

```

0166 CA8B01      JP   Z,OUTTET
0169 FE77       CP   PG5KEY
0168 CAAE01     JP   Z,HSRQ2
016E FE6F       CP   PG6KEY
0170 CA9803     JP   Z,DATOUT
0173 FE5F       CP   PG7KEY
0175 CA8E01     JP   Z,DATA C
0178 FE3F       CP   PG8KEY
017A CA9E01     JP   Z,KAIKOU
017D C31301     JP   MAIN

```

```

;
; JUMP TABLE
;

```

```

0180 CD7A05     AUTO2: CALL SAITAN
0183 0E40       LD   C,NEUT
0185 CD8600     CALL PWSHFT
0188 C31301     JP   MAIN
018B C3F601     OUTTET: JP   OUTTST
018E 0E20       DATA C: LD   C,FIRST
0190 CD8600     CALL PWSHFT
0193 CD9E02     CALL DATA C
0196 0E40       LD   C,NEUT
0198 CD8600     CALL PWSHFT
019B C31301     JP   MAIN
019E 0E20       KAIKOU: LD  C,FIRST
01A0 CD8600     CALL PWSHFT
01A3 CDE803     CALL KAIKO
01A6 0E40       LD   C,NEUT
01A8 CD8600     CALL PWSHFT
01AB C31301     JP   MAIN

```

```

;
;

```

```

01AE 3E02       HSRQ2: LD   A,02H           ;REQUEST AD DATA TO SUBCPU
01B0 D303       OUT  (03H),A
01B2 DB01       HSR1:  IN   A,(01H)       ;NOT START KEY ?
01B4 FEFF       CP   OFFH
01B6 20FA       JR   NZ,HSR1
01B8 3E03       LD   A,03H           ;DRQ2=HI
01BA D303       OUT  (03H),A
01BC CD2400     HSR2:  CALL KEYINP       ;STOP KEY ?
01BF FEFD       CP   STPKEY
01C1 20F9       JR   NZ,HSR2
01C3 CDE900     CALL HSRQ3           ;SUBCPU RESET
01C6 C31301     JP   MAIN

```

```

;
; CUTTER UP SUBROUTINE
;

```

```

01C9 3E05       CUTUP: LD   A,UPON           ;UPPER SOLENOID ON
01CB D383       OUT  (83H),A
01CD DB81       CUTUP1: IN  A,(81H)       ;UP LIMIT?
01CF CB4F       BIT  1,A
01D1 28FA       JR   Z,CUTUP1
01D3 3E04       LD   A,UPOF           ;UPPER SOLENOID OFF
01D5 D383       OUT  (83H),A
01D7 C9        RET

```

```

;
; CUTTER DOWN SUBROUTINE
;

```

```

01D8 3E07       CUTDW: LD  A,DWNON
01DA D383       OUT  (83H),A

```



```

01DC D881      CUTDW1:IN  A,(81H)
01DE C857          BIT  2,A
01E0 28FA          JR   Z,CUTDW1
01E2 3E06          LD   A,DWNOF
01E4 D383          OUT  (83H),A
01E6 C9           RET

;
;HSSR COMPARE SUBROUTINE
; HL=COMPARE VALUE
; EX. 45.0°=0450H
;      -45.0°=8450H
;
01E7 CDB000      HCOMP: CALL HSSR          ;CALL HAND SHAKE SUBROUTINE
01EA ED5B0C30    LD   DE,(SDATA)          ;DE=TURN ANGLE
01EE AF          XOR  A              ;CLEAR CARRY FLAG
01EF E5          PUSH HL
01F0 ED52          SBC  HL,DE
01F2 E1          POP  HL
01F3 30F2          JR   NC,HCOMP
01F5 C9           RET

;
;OUTPUT TEST ROUTINE
;
0001      RGTON: EQU  01H          ;RIGHT ON
0000      RGTOF: EQU  00H          ;RIGHT OFF
0003      LFTON: EQU  03H          ;LEFT ON
0002      LFTOF: EQU  02H          ;LEFT OFF
0005      UPON:  EQU  05H          ;CUTTER UP ON
0004      UPOF:  EQU  04H          ;CUTTER UP OFF
0007      DWNON: EQU  07H          ;CUTTER DOWN ON
0006      DWNOF: EQU  06H          ;CUTTER DOWN OFF
0009      ACCON: EQU  09H          ;PWS ACCELERATE ON
0008      ACCOF: EQU  08H          ;PWS ACCELERATE OFF
000B      DECON: EQU  0BH          ;PWS DECELERATE ON
000A      DECOF: EQU  0AH          ;PWS DECELERATE OFF
;
01F6 3E01      OUTTST:LD  A,RGTON          ;TURN RIGHT 2 SEC
01F8 D383          OUT  (83H),A
01FA CD7402      CALL TIMER1
01FD 3E00          LD   A,RGTOF
01FF D383          OUT  (83H),A
0201 CD7402      CALL TIMER1
0204 3E03          LD   A,LFTON          ;TURN LEFT 2 SEC
0206 D383          OUT  (83H),A
0208 CD7402      CALL TIMER1
020B 3E02          LD   A,LFTOF
020D D383          OUT  (83H),A
020F CD7402      CALL TIMER1
0212 CDD801      CALL CUTDW          ;CUTTER DOWN
0215 CD7402      CALL TIMER1
0218 CDC901      CALL CUTUP          ;CUTTER UP
021B CD7402      CALL TIMER1
021E CDD801      CALL CUTDW          ;CUTTER DOWN
0221 CD7402      CALL TIMER1
0224 0E20          LD   C,FIRST          ;FIRST SPEED
0226 CD8600      CALL PWSHFT
0229 CD7402      CALL TIMER1
022C 0E10          LD   C,SECOND          ;SECOND SPEED
022E CD8600      CALL PWSHFT
0231 CD7402      CALL TIMER1

```

```

0234 0E08          LD   C,THIRD          ;THIRD SPEED
0236 CD8600        CALL PWSHFT
0239 CD7402        CALL TIMER1
023C 0E10          LD   C,SECOND        ;SECOND SPEED
023E CD8600        CALL PWSHFT
0241 CD7402        CALL TIMER1
0244 0E20          LD   C,FIRST          ;FIRST SPEED
0246 CD8600        CALL PWSHFT
0249 CD7402        CALL TIMER1
024C 0E40          LD   C,NEUT          ;NEUTRAL
024E CD8600        CALL PWSHFT
0251 CD7402        CALL TIMER1
0254 0E80          LD   C,REVRS          ;REVERSE
0256 CD8600        CALL PWSHFT
0259 CD7402        CALL TIMER1
025C 0E40          LD   C,NEUT          ;NEUTRAL
025E CD8600        CALL PWSHFT
0261 CD7402        CALL TIMER1
0264 0E08          LD   C,THIRD          ;THIRD
0266 CD8600        CALL PWSHFT
0269 CD7402        CALL TIMER1
026C 0E40          LD   C,NEUT          ;NEUTRAL
026E CD8600        CALL PWSHFT
0271 C31301        JP   MAIN

;
;TIMER SUBROUTINE
;
0274 C5            TIMER1:PUSH BC
0275 0628          LD   B,28H          ;2 SEC TIMER
0277 1817          JR   TIM
0279 C5            TIMER2:PUSH BC
027A 0614          LD   B,14H          ;1 SEC TIMER
027C 1812          JR   TIM
027E C5            TIMER3:PUSH BC
027F 0608          LD   B,08H          ;.4 SEC TIMER
0281 180D          JR   TIM
0283 C5            TIMER4:PUSH BC
0284 0604          LD   B,04H          ;.2 SEC TIMER
0286 1808          JR   TIM
0288 C5            TIMER5:PUSH BC
0289 0602          LD   B,02H          ;.1 SEC TIMER
028B 1803          JR   TIM
028D C5            TIMER6:PUSH BC
028E 0601          LD   B,01H          ;.05 SEC TIMER
0290 D5            TIM:  PUSH DE
0291 11081E        TIM1: LD   DE,1E08H
0294 1B            TIM2: DEC  DE
0295 7A            LD   A,D
0296 B3            OR   E
0297 20FB          JR   NZ,TIM2
0299 10F6          DJNZ TIM1
029B D1            POP  DE
029C C1            POP  BC
029D C9            RET

;
;DATA ACQUISITION SUBROUTINE
;
029E 210000        DATACQ:LD   HL,0
02A1 221C30        LD   (ANGLE),HL      ;CLEAR ANGLE DATA
02A4 220030        LD   (TIMCNT),HL     ;CLEAR TIME COUNTER

```

```

02A7 002A0630      LD      IX,(DATADR)      ;SET DATA AERA TOP ADDRESS
02AB 0600          LD      B,00H           ;DATA NUMBER CLEAR
02AD 3E07          LD      A,07H           ;TIMER START
02AF 0303          OUT     (03H),A
02B1 0880          DATAC0:IN  A,(80H)      ;SEARCH PADDY RICE
02B3 FE09          CP      09H
02B5 28FA          JR      Z,DATAC0
02B7 0880          DATAC1:IN  A,(80H)      ;SEARCH PADDY RICE
02B9 FE09          CP      09H
02BB 203B          JR      NZ,DATAC2
02BD 007902        CALL   TIMER2           ;1 SEC DELAY
02C0 0880          IN      A,(80H)       ;INPUT AGAIN
02C2 FE09          CP      09H
02C4 2032          JR      NZ,DATAC2
02C6 3E06          LD      A,06H           ;TIMER STOP
02C8 0303          OUT     (03H),A
02CA 003403        CALL   TRDATM           ;STORE LAST DATA
02CD 006603        CALL   TRDAAG
02D0 04           INC      B
02D1 3EFF          LD      A,0FFH         ;DATA TERMINATER=FFH
02D3 007700        LD      (IX),A         ;WRITE TERMINATER 3BYTE
02D6 0023          INC     IX
02D8 007700        LD      (IX),A
02DB 0023          INC     IX
02DD 007700        LD      (IX),A
02E0 2A0430        LD      HL,(DACTAD)    ;WRITE DATA NUMBER
02E3 70           LD      (HL),B
02E4 0023          INC     IX
02E6 00220430      LD      (DACTAD),IX    ;SET NEXT DATA COUNTER ADDRESS
02EA 0023          INC     IX
02EC 00220630      LD      (DATAQR),IX    ;SET NEXT DATA TOP ADDRESS
02F0 3A0010        LD      A,(DATNUM)
02F3 3C           INC     A
02F4 320010        LD      (DATNUM),A
02F7 C9           RET
02F8 0880          DATAC2:IN  A,(80H)      ;SEARCH PADDY RICE
02FA C86F          BIT     5,A            ;INTERNAL LEFT SENSOR ?
02FC 2006          JR      NZ,DATAC3
02FE C87F          BIT     7,A            ;INTERNAL RIGHT SENSOR ?
0300 201A          JR      NZ,DATAC4
0302 18B3          JR      DATAC1
0304 003403        DATAC3:CALL TRDATM     ;SET TIME DATA
0307 00E900        CALL   HSRQ3           ;RESET SUBCPU
030A 007A03        DAC1:  CALL  LFTTRN     ;TURN LEFT (PULSE)
030D 0880          IN      A,(80H)
030F C86F          BIT     5,A            ;INTERNAL LEFT SENSOR OFF
0311 20F7          JR      NZ,DAC1
0313 00B000        CALL   HSSR           ;GET TURN ANGLE
0316 006603        CALL   TRDAAG        ;SET TURN ANGLE DATA
0319 04           INC     B              ;DATA NUMBER INCREMENT
031A 189B          JR      DATAC1
031C 003403        DATAC4:CALL TRDATM
031F 00E900        CALL   HSRQ3
0322 008903        DAC2:  CALL  RGTTRN
0325 0880          IN      A,(80H)
0327 C87F          BIT     7,A
0329 20F7          JR      NZ,DAC2
032B 00B000        CALL   HSSR
032E 006603        CALL   TRDAAG
0331 04           INC     B

```

```

0332 1883          JR    DATA1
;
0334 78           TRDATM:LD  A,B           ;FIRST DATA ?
0335 FE00         CP    00H
0337 281C         JR    Z,TRD1
0339 2A0030       LD    HL,(TIMCNT)       ;DATA=(TIME COUNTER)-(TIME BUFFER)
033C ED5B0230     LD    DE,(TIMBUF)
0340 AF           XOR   A
0341 7D           LD    A,L
0342 9B           SBC   A,E
0343 27           DAA
0344 D07700       LD    (IX),A
0347 7C           LD    A,H
0348 9A           SBC   A,D
0349 27           DAA
034A D023         INC   IX
034C D07700       LD    (IX),A
034F D023         INC   IX
0351 220230       LD    (TIMBUF),HL
0354 C9           RET
;
0355 2A0030       TRD1: LD    HL,(TIMCNT)
0358 D07500       LD    (IX),L
035B D023         INC   IX
035D D07400       LD    (IX),H
0360 D023         INC   IX
0362 220230       LD    (TIMBUF),HL
0365 C9           RET
;
0366 2A1C30       TRDAAG:LD  HL,(ANGLE)       ;SET TURN ANGLE DATA
0369 D07500       LD    (IX),L
036C D023         INC   IX
036E D07400       LD    (IX),H
0371 D023         INC   IX
0373 2A0C30       LD    HL,(SDATA)
0376 221C30       LD    (ANGLE),HL
0379 C9           RET
;
037A 3E03         LFTTRN:LD  A,LFTON       ;PULSE WIDTH CONTROL
037C D383         OUT   (83H),A
037E CD8302       CALL  TIMER4       ;.2 SEC ON
0381 3E02         LD    A,LFTOF
0383 D383         OUT   (83H),A
0385 CD8802       CALL  TIMER5       ;.1 SEC OFF
0388 C9           RET
;
0389 3E01         RGTRN:LD  A,RGTON
038B D383         OUT   (83H),A
038D CD8302       CALL  TIMER4
0390 3E00         LD    A,RGTOF
0392 D383         OUT   (83H),A
0394 CD8802       CALL  TIMER5
0397 C9           RET
;
;DATA OUTPUT ROUTINE
;
0398 3E83         DATOUT:LD  A,83H           ;SET DATA OUTPUT MODE 8255 #0
039A D303         OUT   (03H),A
039C D802         DAOUT1:IN  A,(02H)
039E CB47         BIT   0,A           ;DATA REQUEST?

```

```

03A0 20FA          JR    NZ,DAOUT1
03A2 210010       LD    HL,DATNUM          ;SEND DATA ADDRESS SET
03A5 DB02         DAOUT2:IN  A,(02H)
03A7 CB47         BIT    0,A
03A9 280B         JR    Z,DAOUT3
03AB 3E9A         LD    A,9AH          ;SET NORMAL MODE 8255 #0
03AD D303         OUT   (03H),A
03AF 3E07         LD    A,07H
03B1 D302         OUT   (02H),A
03B3 C31301      JP    MAIN
03B6 3E0E         DAOUT3:LD  A,0EH          ;BUSY=LOW
03B8 D303         OUT   (03H),A
03BA 7E           LD    A,(HL)          ;LSB DATA SET
03BB D300         OUT   (00H),A
03BD 3E0C         LD    A,0CH          ;STROBE=LOW
03BF D303         OUT   (03H),A
03C1 DB02         DAOUT4:IN  A,(02H)      ;DRQ=HI?
03C3 CB47         BIT    0,A
03C5 28FA         JR    Z,DAOUT4
03C7 3E0D         LD    A,0DH          ;STROBE=HI
03C9 D303         OUT   (03H),A
03CB 23           INC   HL
03CC 7E           LD    A,(HL)          ;MSB DATA SET
03CD D300         OUT   (00H),A
03CF 23           INC   HL
03D0 3E0C         LD    A,0CH          ;STROBE=LOW
03D2 D303         OUT   (03H),A
03D4 DB02         DAOUT5:IN  A,(02H)      ;DRQ=LOW?
03D6 CB47         BIT    0,A
03D8 20FA         JR    NZ,DAOUT5
03DA 3E0D         LD    A,0DH          ;STROBE=HI
03DC D303         OUT   (03H),A
03DE 3E0F         LD    A,0FH          ;BUSY=HI
03E0 D303         OUT   (03H),A
03E2 0600         LD    B,00H          ;DELAY FOR DRQ DETECT
03E4 10FE         DJNZ  $
03E6 18B0         JR    DAOUT2

;
;KAIKO SUBROUTINE
;
03E8 CDC901      KAIKO: CALL CUTUP          ;CUTTER UP
03EB CDE900      CALL HSRQ3          ;RESET SUBCPU
03EE 3E03         LD    A,LFTON          ;TURN LEFT
03F0 D383         OUT   (83H),A
03F2 215085      LD    HL,8550H          ;-55.0° TURN
03F5 CDE701      CALL HCOMP
03F8 3E02         LD    A,LFTOF
03FA D383         OUT   (83H),A
03FC 0E80         LD    C,REVRS          ;BACKWARD
03FE CD8600      CALL PWSHFT
0401 3A0830      LD    A,(BATMDA)      ;SET BACK TIME DATA
0404 47           LD    B,A
0405 CD8D02      KAIKO1:CALL TIMER6
0408 DB80         IN    A,(80H)          ;INPUT OUTER LEFT SENSOR
040A CB77         BIT    6,A
040C 2002         JR    NZ,KAIKO2      ;DETECTED CORNER ?
040E 10F5         DJNZ  KAIKO1          ;TIME OVER ?
0410 CD7E02      KAIKO2:CALL TIMER3      ;.4 SEC TIMER
0413 CDE900      CALL HSRQ3          ;RESET SUBCPU
0416 3E01         LD    A,RGTON          ;TURN RIGHT

```

```

0418 D383          OUT (83H),A
041A 215082       LD HL,8250H          ;-25.0° TURN
041D CDE701       CALL HCOMP
0420 3E00         LD A,RGTOF
0422 D383          OUT (83H),A
0424 CDD801       CALL CUTDW          ;CUTTER DOWN
0427 0E20         LD C,FIRST          ;FORWARD AT THE FIRST SPEED
0429 CD8600       CALL PWSHFT
042C 06A0         LD B,0A0H          ;TIME DATA ABOUT 8 SEC
042E CD8D02       KAIK03:CALL TIMER6   ;.05 SEC TIMER
0431 D881         IN A,(81H)          ;INPUT OUTER RIGHT SENSOR
0433 CB47         BIT 0,A
0435 280E         JR Z,KAIK04
0437 CD7E02       CALL TIMER3          ;WAIT .4 SEC
043A D881         IN A,(81H)          ;INPUT AGAIN
043C CB47         BIT 0,A
043E 2805         JR Z,KAIK04
0440 CD5704       CALL RGTPAL
0443 18E9         JR KAIK03
0445 D880         KAIK04:IN A,(80H)    ;INPUT INTERNAL RIGHT SENSOR
0447 CB5F         BIT 3,A
0449 280A         JR Z,KAIK05
044B 10E1         DJNZ KAIK03        ;DETECTED PADDY RICE ?
044D 0E40         LD C,NEUT          ;TIME OVER ?
044F CD8600       CALL PWSHFT        ;STOP COMBINE
0452 3EFF         LD A,OFFH          ;STOP FLAG ON
0454 C9           RET
0455 AF           KAIK05:XOR A        ;STOP FLAG OFF
0456 C9           RET

;
;RIGHT PARALLEL TURN
;
0457 0E80         RGTPAL:LD C,REVRS   ;BACKWARD
0459 CD8600       CALL PWSHFT
045C CDC901       CALL CUTUP          ;CUTTER UP
045F CDE900       CALL HSRQ3          ;SUBCPU RESET
0462 3E01         LD A,RGTON          ;TURN RIGHT
0464 D383          OUT (83H),A
0466 215084       LD HL,8450H          ;-45.0° TURN
0469 CDE701       CALL HCOMP
046C 3E00         LD A,RGTOF
046E D383          OUT (83H),A
0470 CD7902       CALL TIMER2          ;1 SEC (ABOUT 35cm)
0473 CDE900       CALL HSRQ3
0476 3E03         LD A,LFTON          ;TURN LEFT
0478 D383          OUT (83H),A
047A 215004       LD HL,0450H          ;45.0° TURN
047D CDE701       CALL HCOMP
0480 3E02         LD A,LFTOF
0482 D383          OUT (83H),A
0484 0E20         LD C,FIRST          ;FORWARD AT THE FIRST SPEED
0486 CD8600       CALL PWSHFT
0489 CDD801       CALL CUTDW          ;CUTTER DOWN
048C C9           RET

;
;NORMAL AUTOMATIC TRAVELING
;
048D 0E20         AUTO1: LD C,FIRST   ;FORWARD AT THE FIRST SPEED
048F CD8600       CALL PWSHFT
0492 D880         AUT1: IN A,(80H)   ;SEARCH PADDY RICE?

```

```

0494 FE09          CP    09H
0496 28FA          JR    Z,AUT1
0498 DB81          AUT2: IN    A,(81H)          ;INPUT OUTER RIGHT SENSOR
049A CB47          BIT    0,A
049C 280E          JR    Z,AUT8          ;DETECTED PADDY RICE IN RIGHT SIDE
049E CD7E02        CALL  TIMER3          ;WAIT .4 SEC
04A1 DB81          IN    A,(81H)          ;AGAIN
04A3 CB47          BIT    0,A
04A5 2805          JR    Z,AUT8
04A7 CD5704        CALL  RGTPAL
04AA 18EC          JR    AUT2
04AC 3A3C30        AUT8: LD    A,(TRFLG)
04AF FE00          CP    0
04B1 2828          JR    Z,AUT6
04B3 1800          JR    AUT3

;
04B5 DB80          AUT3: IN    A,(80H)          ;INPUT INTERNAL RIGHT SENSOR
04B7 CB67          BIT    4,A
04B9 280C          JR    Z,AUT4          ;DETECTED PADDY RICE
04B8 CD7E02        CALL  TIMER3          ;WAIT .4 SEC
04BE DB80          IN    A,(80H)          ;AGAIN
04C0 CB67          BIT    4,A
04C2 2803          JR    Z,AUT4
04C4 CD8903        CALL  RGTTRN          ;RIGHT TURN
04C7 DB80          AUT4: IN    A,(80H)          ;INPUT INTERNAL RIGHT SENSOR
04C9 CB5F          BIT    3,A
04CB 283A          JR    Z,AUT5
04CD CD7E02        CALL  TIMER3          ;WAIT .4 SEC
04D0 DB80          IN    A,(80H)          ;INPUT AGAIN
04D2 CB5F          BIT    3,A
04D4 2831          JR    Z,AUT5
04D6 CD7A03        CALL  LFTTRN          ;LEFT TURN
04D9 182C          JR    AUT5

;
04DB DB80          AUT6: IN    A,(80H)
04DD E607          AND    7H
04DF FE05          CP    5H
04E1 2806          JR    Z,AUT7
04E3 FE06          CP    6H
04E5 2812          JR    Z,AUT11
04E7 181E          JR    AUT5
04E9 CD8302        AUT7: CALL  TIMER4
04EC DB80          IN    A,(80H)
04EE E607          AND    7H
04F0 FE05          CP    5H
04F2 2013          JR    NZ,AUT5
04F4 CD8903        CALL  RGTTRN
04F7 180E          JR    AUT5
04F9 CD8302        AUT11: CALL  TIMER4
04FC DB80          IN    A,(80H)
04FE E607          AND    7H
0500 FE06          CP    6H
0502 2003          JR    NZ,AUT5
0504 CD7A03        CALL  LFTTRN

;
0507 DB80          AUT5: IN    A,(80H)          ;INPUT ALL STEERING SENSORS
0509 FE09          CP    09H
050B 208B          JR    NZ,AUT2
050D CD7E02        CALL  TIMER3          ;WAIT .7 SEC
0510 CD8302        CALL  TIMER4

```

```

0513 CD8802          CALL  TIMER5
0516 D880           IN    A,(80H)
0518 FE09           CP    09H
051A C29804         JP    NZ,AUT2
051D C0E803         CALL  KAIKO
0520 FEFF           CP    OFFH          ;STOP FLAG ON ?
0522 CA1301         JP    Z,MAIN
0525 3A3C30         LD    A,(TRFLG)
0528 FE00           CP    0
052A 2804           JR    Z,AUT9
052C 3E00           LD    A,0
052E 1802           JR    AUT10
0530 3E01           AUT9: LD    A,1
0532 323C30         AUT10: LD   (TRFLG),A
0535 C39804         JP    AUT2

;
;BACK TIME DATA CHANGE ROUTINE
;
0538 CD2400         BKTIME:CALL KEYINP
0538 FEF7           CP    PG1KEY          ;P-1 KEY ?
053D 280B           JR    Z,BKTIM1
053F FEDF           CP    PG3KEY          ;P-3 KEY ?
0541 281F           JR    Z,BKTIM2
0543 FEFD           CP    STPKEY          ;STOP KEY ?
0545 CA1301         JP    Z,MAIN          ;GOTO MAIN
0548 18EE           JR    BKTIME
054A 3E0D           BKTIM1:LD   A,0DH          ;TPF LAMP ON
054C D383           OUT   (83H),A
054E CD7E02         CALL  TIMER3
0551 3E0C           LD    A,0CH          ;TPF LAMP OFF
0553 D383           OUT   (83H),A
0555 CD7E02         CALL  TIMER3
0558 3A0830         LD    A,(BATMDA)      ;BACK TIME DATA + 0.5 SEC
0558 C60A           ADD   A,0AH
055D 320830         LD    (BATMDA),A
0560 18D6           JR    BKTIME
0562 3E0F           BKTIM2:LD   A,0FH          ;BRK LAMP ON
0564 D383           OUT   (83H),A
0566 CD7E02         CALL  TIMER3
0569 3E0E           LD    A,0EH          ;BRK LAMP OFF
056B D383           OUT   (83H),A
056D CD7E02         CALL  TIMER3
0570 3A0830         LD    A,(BATMDA)      ;BACK TIME DATA - 0.5 SEC
0573 D60A           SUB   0AH
0575 320830         LD    (BATMDA),A
0578 18BE           JR    BKTIME

;
;
*****
*                               *
*   SAITAN KYORI ENZAN         *
*                               *
*   MAIN ROUTINE               *
*                               *
*****
;
;SAITAN:CALL CTAR          ;GET X-Y DATA
057A CD1706         CALL  SLV          ;SEARCH LIMIT VALUE
057D CD4409         CALL  COTN         ;CALCULATE SAITAN ANGLE & TIME
0580 CD370A

```



```

;
0583 210000      LD   HL,0           ;CLEAR TIME COUNTER
0586 220030      LD   (TIMCNT),HL
;
0589 0E20        LD   C,FIRST       ;FORWARD FIRST SPEED
058B CD8600      CALL PWSHFT
058E 3E07        LD   A,07H        ;TIMER START
0590 D303        OUT  (03H),A
;
0592 DD210016    LD   IX,ATOBJD      ;SET OBJECTIVE DATA BUFFER
0596 ED5B0210    LD   DE,(DATBUF)      ;SET FIRST OBJECTIVE TIME
059A ED531230    LD   (TIMS),DE
;
059E AF          SAIT1: XOR  A           ;CLEAR CARRY FLAG
059F 2A0030      LD   HL,(TIMCNT)    ;IF (DATBUF)<<(TIMCNT) THEN NEXT EL
05A2 ED52        SBC  HL,DE
05A4 38F8        JR   C,SAIT1
;
05A6 DD6E00      SAIT2: LD   L,(IX)      ;HL<---OBJECTIVE FIRST DATA
05A9 DD23        INC  IX
05AB DD6600      LD   H,(IX)
05AE DD23        INC  IX
05B0 7D          LD   A,L           ;IF (HL)=FFFF THEN END ELSE NEXT
05B1 FEFF        CP   0FFH
05B3 2840        JR   Z,SAIEND
05B5 B4          OR   H           ;IF (HL)=0000 THEN NOT CALUCULATE
05B6 281C        JR   Z,SAIT6
;
05B8 CDE900      CALL HSRQ3          ;RESET SUB CPU
05BB CB7C        BIT  7,H           ;RIGHT OR LEFT DATA ?
05BD 2806        JR   Z,SAIT4
05BF 3E03        LD   A,LFTON
05C1 D383        OUT  (83H),A
05C3 1804        JR   SAI5
05C5 3E01        SAIT4: LD   A,RGTON
05C7 D383        OUT  (83H),A
05C9 CDE701      SAIT5: CALL HCOMP
05CC 3E02        LD   A,LFTOF
05CE D383        OUT  (83H),A
05D0 3E00        LD   A,RGTOF
05D2 D383        OUT  (83H),A
;
05D4 2A1230      SAIT6: LD   HL,(TIMS)    ;(TIMS)=(TIMS)+(IX)
05D7 7D          LD   A,L
05D8 DD8600      ADD  A,(IX)
05DB 27          DAA
05DC 5F          LD   E,A
05DE DD23        INC  IX
05DF 7C          LD   A,H
05E0 DD8E00      ADC  A,(IX)
05E3 27          DAA
05E4 57          LD   D,A
05E5 DD23        INC  IX
05E7 ED531230    LD   (TIMS),DE
05EB AF          SAIT7: XOR  A           ;IF (TIMCNT)>>(TIMS) THEN NEXT ELSE
05EC 2A0030      LD   HL,(TIMCNT)
05EF ED52        SBC  HL,DE
05F1 38F8        JR   C,SAIT7
05F3 18B1        JR   SAIT2
;

```

```

05F5 DB80      SAIEND:IN  A,(80H)          ;SEARCH PADDY RICE
05F7 FE09      CP      09H
05F9 200E      JR      NZ,SAIEN1
05FB CD7902    CALL  TIMER2          ;WAIT 1 SEC
05FE DB80      IN      A,(80H)
0600 FE09      CP      09H
0602 2005      JR      NZ,SAIEN1
0604 3E06      LD      A,06H          ;TIMER STOP
0606 D303      OUT    (03H),A
0608 C9        RET

;
0609 DB80      SAIEN1:IN  A,(80H)
060B CB6F      BIT    5,A
060D C47A03    CALL  NZ,LFTTRN
0610 CB7F      BIT    7,A
0612 C48903    CALL  NZ,RGTTTRN
0615 180E      JR      SAIEND

;
*****
*
*  SAITAN KYORI ENZAN *
*
*  SUB ROUTINE *
*
*****
;
0617 210110    CTAR: LD    HL,DATCNT      ;SET DATA NUMBER TO B reg.
061A 46        LD    B,(HL)
061B 78        LD    A,B
061C 320012    LD    (XYIBUF),A          ;STORE DATA NUMBER
061F DD210112  LD    IX,XYIBUF+1          ;SET RESULT DATA ADDRESS
0623 210000    LD    HL,0000H          ;CLEAR XID,YID,ANGS
0626 221830    LD    (XID),HL
0629 221A30    LD    (YID),HL
062C 221630    LD    (ANGS),HL
062F 210210    LD    HL,DATBUF          ;SET BCD DATA TOP ADDRESS
0632 C5        CTAR1: PUSH BC          ;SAVE LOOP COUNTER
0633 E5        PUSH HL          ;SAVE BCD TIME DATA ADDRESS
0634 CD7507    CALL  ZDB2B          ;TIME DATA CONVERT TO BINARY DATA
0637 211430    LD    HL,TIMB          ;STORE TO TIMB
063A CD9A06    CALL  ZST2B
063D E1        POP  HL          ;LOAD BCD DATA ADDRESS
063E 23        INC  HL          ;SET BCD ANGLE ADDRESS
063F 23        INC  HL
0640 CD7507    CALL  ZDB2B          ;ANGLE DATA CONVERT TO BINARY DATA
0643 E5        PUSH HL          ;SAVE BCD DATA ADDRESS
0644 211030    LD    HL,ANG1          ;STORE TO ANG1
0647 CD9A06    CALL  ZST2B
064A CD9506    CALL  ZLD2B          ;ANGS=ANG1+ANGS
064D 211630    LD    HL,ANGS
0650 CD9F06    CALL  ZAD2B
0653 211630    LD    HL,ANGS
0656 CD9A06    CALL  ZST2B
0659 CD1D08    CALL  COS          ;COS(ANGS)
065C 211430    LD    HL,TIMB          ;TIMB*COS(ANGS)
065F CDCF06    CALL  ZML2B
0662 211830    LD    HL,XID          ;XID=TIMB*COS(ANGS)+XID
0665 CD8406    CALL  CTAR2

```

```

0668 211630      LD   HL,ANGS      ;SIN(ANGS)
066B CD9506      CALL ZLD2B
066E CDA708      CALL SIN
0671 211430      LD   HL,TIMB      ;TIMB*SIN(ANGS)
0674 CDCF06      CALL ZML2B
0677 211A30      LD   HL,YID      ;YID=TIMB*SIN(ANGS)+YID
067A CD8406      CALL CTAR2
067D E1          POP  HL
067E 23          INC  HL
067F 23          INC  HL
0680 C1          POP  BC
0681 10AF        DJNZ CTAR1
0683 C9          RET

```

```

;
0684 CD9F06      CTAR2: CALL ZAD2B
0687 CD9A06      CALL ZST2B
068A DD7300      LD   (IX),E
068D DD23        INC  IX
068F DD7200      LD   (IX),D
0692 DD23        INC  IX
0694 C9          RET

```

```

;
*****
*
* 2 BYTES CALCULATING SUBROUTINE *
*
*****

```

```

;
;LOAD 2 BYTES
;
;(HL+1,HL)---->D,E
;
0695 5E          ZLD2B: LD   E,(HL)
0696 23          INC  HL
0697 56          LD   D,(HL)
0698 2B          DEC  HL
0699 C9          RET

```

```

;STORE 2 BYTES
;
;D,E---->(HL+1,HL)
;
069A 73          ZST2B: LD   (HL),E
069B 23          INC  HL
069C 72          LD   (HL),D
069D 2B          DEC  HL
069E C9          RET

```

```

;ADD 2 BYTE
;
;D,E+(HL+1,HL)---->D,E
;
069F E5          ZAD2B: PUSH HL
06A0 7E          LD   A,(HL)
06A1 83          ADD  A,E
06A2 5F          LD   E,A
06A3 23          INC  HL
06A4 7E          LD   A,(HL)

```

```

06A5 8A          ADC  A,D
06A6 57          LD   D,A
06A7 2B          DEC  HL
06A8 E1          POP  HL
06A9 C9          RET

;
;SUBTRACT 2 BYTES
;
;D,E-(HL+1,HL)---->D,E
;
06AA E5          ZSB2B: PUSH HL
06AB 7B          LD   A,E
06AC 96          SUB  (HL)
06AD 5F          LD   E,A
06AE 23          INC  HL
06AF 7A          LD   A,D
06B0 9E          SBC  A,(HL)
06B1 57          LD   D,A
06B2 2B          DEC  HL
06B3 E1          POP  HL
06B4 C9          RET

;
;COMPARE 2 BYTES
;
;DE-(HL+1,HL) =ZF, <CY
;
06B5 D5          ZCP2B: PUSH DE
06B6 CDAA06      CALL ZSB2B
06B9 3806        JR   C,ZCP2B1
06BB 7A          LD   A,D
06BC B3          OR   E
06BD 2802        JR   Z,ZCP2B1
06BF AF          XOR  A
06C0 3C          INC  A
06C1 D1          ZCP2B1:POP DE
06C2 C9          RET

;
;MULTIPLY 10 TO 2 BYTES
;
;DE * 10---->DE
;
06C3 E5          ZM12B: PUSH HL
06C4 62          LD   H,D
06C5 68          LD   L,E
06C6 29          ADD  HL,HL
06C7 54          LD   D,H
06C8 5D          LD   E,L
06C9 29          ADD  HL,HL
06CA 29          ADD  HL,HL
06CB 19          ADD  HL,DE
06CC EB          EX  DE,HL
06CD E1          POP  HL
06CE C9          RET

;
;MULTIPLY 2 BYTES
;
;D,E * (HL+1,HL)---->B,C,D,E
;
06CF 010000      ZML2B: LD   BC,0
06D2 CDFD07      CALL ZAB2B

```

```

06D5 3E00          LD  A,0
06D7 3001          JR  NC,ZML1
06D9 3C           INC  A
06DA F5           ZML1: PUSH AF
06DB D5           PUSH DE
06DC CD9506       CALL ZLD2B
06DF CDFD07       CALL ZAB2B
06E2 CD9A06       CALL ZST2B
06E5 D1           POP  DE
06E6 3803         JR  C,ZML3
06E8 F1           POP  AF
06E9 1802         JR  ZML2
06EB F1           ZML3: POP  AF
06EC 3C           INC  A
06ED F5           ZML2: PUSH AF
06EE E5           PUSH HL
06EF 2610         LD   H,16

;
06F1 E3           ZML2B1:EX  (SP),HL
06F2 CB1A         RR   D
06F4 CB1B         RR   E
06F6 3008         JR   NC,ZML2B2

;
06F8 79           LD   A,C
06F9 86           ADD  A,(HL)
06FA 4F           LD   C,A
06FB 23           INC  HL
06FC 78           LD   A,B
06FD 8E           ADC  A,(HL)
06FE 47           LD   B,A
06FF 2B           DEC  HL

;
0700 CB18         ZML2B2:RR  B
0702 CB19         RR  C

;
0704 E3           EX  (SP),HL
0705 25           DEC  H
0706 20E9         JR  NZ,ZML2B1

;
0708 CB1A         RR  D
070A CB1B         RR  E

;
070C E1           POP  HL
070D F1           POP  AF
070E FE01         CP   1H
0710 C0           RET  NZ
0711 CD0508       CALL ZSN4B
0714 C9           RET

;
;DIVIDE 4 BYTES BY 2 BYTES
;
;B,C,D,E/(HL+1,HL)---->D,E
;
0715 CD1508       ZDV2B: CALL ZAB4B
0718 3E00         LD   A,0
071A 3001         JR   NC,ZDV1
071C 3C           INC  A
071D F5           ZDV1: PUSH AF
071E D5           PUSH DE
071F CD9506       CALL ZLD2B

```

```

0722 CDFD07      CALL ZAB2B
0725 CD9A06      CALL ZST2B
0728 D1          POP DE
0729 3803        JR C,ZDV3
072B F1          POP AF
072C 1802        JR ZDV2
072E F1          ZDV3: POP AF
072F 3C          INC A
0730 F5          ZDV2: PUSH AF
0731 E5          PUSH HL
0732 2610        LD H,16
0734 E3          ZDV2B1:EX (SP),HL
0735 CB13        RL E
0737 CB12        RL D
0739 CB11        RL C
073B CB10        RL B
;
073D 79          LD A,C
073E 96          SUB (HL)
073F 4F          LD C,A
0740 23          INC HL
0741 78          LD A,B
0742 9E          SBC A,(HL)
0743 47          LD B,A
0744 2B          DEC HL
0745 3009        JR NC,ZDV2B2
;
0747 79          LD A,C
0748 86          ADD A,(HL)
0749 4F          LD C,A
074A 23          INC HL
074B 78          LD A,B
074C 8E          ADC A,(HL)
074D 47          LD B,A
074E 2B          DEC HL
074F 37          SCF
;
0750 3F          ZDV2B2:CCF
0751 E3          EX (SP),HL
0752 25          DEC H
0753 20DF        JR NZ,ZDV2B1
;
0755 CB13        RL E
0757 CB12        RL D
;
0759 E1          POP HL
075A F1          POP AF
075B FE01        CP 1H
075D C0          RET NZ
075E CDF507      CALL ZSN2B
0761 C9          RET
;
;SQUARE ROOT
;
;DE=SQR(DE)
;
0762 E5          ZSQ2B: PUSH HL
0763 EB          EX DE,HL
0764 110000       LD DE,0
0767 37          SCF

```

```

0768 ED52      ZSQ2B1:SBC  HL,DE
076A 3807      JR    C,ZSQ2B2
076C ED52      SBC  HL,DE
076E 3803      JR    C,ZSQ2B2
0770 13        INC  DE
0771 18F5      JR    ZSQ2B1
0773 E1        ZSQ2B2:POP  HL
0774 C9        RET

;
; BCD(40) TO BINARY(2B) BCD(2BYTES)
;
;BCD(HL+1,HL)---->D,E(BINARY) with SIGN
;
0775 E5        ZDB2B: PUSH HL
0776 C5        PUSH BC
0777 CD9506    CALL ZLD2B
077A 7A        LD   A,D
077B 07        RLCA
077C 3E00      LD   A,0
077E 3003      JR   NC,ZDB2B1
0780 CB8A      RES  7,D
0782 3C        INC  A
0783 F5        ZDB2B1:PUSH AF
0784 210E30    LD   HL,BCDDAT
0787 CD9A06    CALL ZST2B
078A 23        INC  HL
078B AF        XOR  A
078C ED6F      RLD
078E 5F        LD   E,A
078F 1600      LD   D,0
0791 ED6F      RLD
0793 47        LD   B,A
0794 CDB007    CALL ZDBSU2
0797 2B        DEC  HL
0798 AF        XOR  A
0799 ED6F      RLD
079B 47        LD   B,A
079C ED6F      RLD
079E 4F        LD   C,A
079F CDB007    CALL ZDBSU2
07A2 41        LD   B,C
07A3 CDB007    CALL ZDBSU2
07A6 F1        POP  AF
07A7 C1        POP  BC
07A8 E1        POP  HL
07A9 CB47      BIT  0,A
07AB C8        RET  Z
07AC CDF507    CALL ZSN2B
07AF C9        RET

;
07B0 CDC306    ZDBSU2:CALL ZM12B
07B3 7B        LD   A,E
07B4 80        ADD  A,B
07B5 5F        LD   E,A
07B6 3E00      LD   A,0
07B8 8A        ADC  A,D
07B9 57        LD   D,A
07BA C9        RET

;
; BINARY 2BYTES TO BCD 4 DIGIT

```

```

;
;D,E(BINARY)---->BCD(HL+1,HL) with SIGN
;
07BB E5      ZBD2B: PUSH HL
07BC C5              PUSH BC
07BD CDFD07        CALL ZAB2B
07C0 3E00          LD  A,0
07C2 3001          JR  NC,ZBD2B1
07C4 3C              INC  A
07C5 F5              ZBD2B1: PUSH AF
07C6 23              INC  HL
07C7 01E803        LD  BC,1000
07CA CDE707        CALL SRT
07CD 016400        LD  BC,100
07D0 CDE707        CALL SRT
07D3 2B              DEC  HL
07D4 010A00        LD  BC,10
07D7 CDE707        CALL SRT
07DA 7B              LD  A,E
07DB ED6F          RLD
07DD F1              POP  AF
07DE C1              POP  BC
07DF E1              POP  HL
07E0 B7              OR   A
07E1 C8              RET  Z
07E2 23              INC  HL
07E3 CBFE          SET  7,(HL)
07E5 2B              DEC  HL
07E6 C9              RET

;
07E7 EB      SRT:  EX  DE,HL
07E8 97              SUB  A
07E9 ED42        SRT1: SBC  HL,BC
07EB 3803        JR   C,SRT2
07ED 3C              INC  A
07EE 18F9        JR   SRT1
07F0 09          SRT2: ADD  HL,BC
07F1 EB          EX  DE,HL
07F2 ED6F        RLD
07F4 C9          RET

;
;SIGN CHANGE 2BYTES
;
;-(D,E)---->D,E
;
07F5 AF      ZSN2B: XOR  A
07F6 93              SUB  E
07F7 5F              LD  E,A
07F8 3E00          LD  A,0
07FA 9A              SBC  A,D
07FB 57              LD  D,A
07FC C9              RET

;
;ABSOLUTE 2BYTES
;
;DE=ABS(DE)
;
07FD 7A          ZAB2B: LD  A,D
07FE 07          RLCA
07FF 00          RET  NC

```



```

0800 CDF507          CALL ZSN2B
0803 37              SCF
0804 C9              RET

;
;SIGN CHANGE 4 BYTES
;
0805 AF          ZSN4B: XOR  A
0806 93          SUB  E
0807 5F          LD   E,A
0808 3E00        LD   A,0
080A 9A          SBC  A,D
080B 57          LD   D,A
080C 3E00        LD   A,0
080E 99          SBC  A,C
080F 4F          LD   C,A
0810 3E00        LD   A,0
0812 98          SBC  A,B
0813 47          LD   B,A
0814 C9          RET

;
;ABSOLUTE 4 BYTES
;
0815 78          ZAB4B: LD   A,B
0816 07          RLCA
0817 D0          RET  NC
0818 CD0508      CALL ZSN4B
081B 37          SCF
081C C9          RET

;
;
;COS
;
; D,E---->D,E=COS(DE)
;
081D E5          COS:  PUSH HL
081E CDF007      CALL ZAB2B
0821 219308      LD   HL,CPDAT
0824 CD8506      CALL ZCP2B
0827 3005        JR   NC,COS1
0829 116400      LD   DE,100
082C E1          POP  HL
082D C9          RET

;
082E CD8008      COS1: CALL COSSUB
0831 3005        JR   NC,COS2
0833 116300      LD   DE,99
0836 E1          POP  HL
0837 C9          RET
0838 CD8008      COS2: CALL COSSUB
083B 3005        JR   NC,COS3
083D 116200      LD   DE,98
0840 E1          POP  HL
0841 C9          RET
0842 CD8008      COS3: CALL COSSUB
0845 3005        JR   NC,COS4
0847 116100      LD   DE,97
084A E1          POP  HL
084B C9          RET
084C CD8008      COS4: CALL COSSUB

```

```

084F 3005          JR   NC,COS5
0851 116000       LD   DE,96
0854 E1           POP  HL
0855 C9           RET
0856 CD8D08      COS5: CALL COSSUB
0859 3005          JR   NC,COS6
085B 115F00       LD   DE,95
085E E1           POP  HL
085F C9           RET
0860 CD8D08      COS6: CALL COSSUB
0863 3005          JR   NC,COS7
0865 115E00       LD   DE,94
0868 E1           POP  HL
0869 C9           RET
086A CD8D08      COS7: CALL COSSUB
086D 3005          JR   NC,COS8
086F 115D00       LD   DE,93
0872 E1           POP  HL
0873 C9           RET
0874 CD8D08      COS8: CALL COSSUB
0877 3005          JR   NC,COS9
0879 115C00       LD   DE,92
087C E1           POP  HL
087D C9           RET
087E CD8D08      COS9: CALL COSSUB
0881 3005          JR   NC,COS10
0883 115B00       LD   DE,91
0886 E1           POP  HL
0887 C9           RET
0888 115A00      COS10: LD   DE,90
088B E1           POP  HL
088C C9           RET

;
088D 23           COSSUB:INC HL
088E 23           INC  HL
088F CDB506       CALL ZCP2B
0892 C9           RET

;
0893 13006600    CPDAT: DW   19,102,130,154,174,192,209,225,239,253
0897 82009A00
089B AE00C000
089F D100E100
08A3 EF00FD00

;
;SIN
;
;D,E--->DE=SIN(DE)
;
08A7 E5           SIN:  PUSH HL
08A8 21CF08       LD   HL,SIND1          ;DE*172
08AB CDCF06       CALL ZML2B
08AE 21D108       LD   HL,SIND2          ;DE*172/100
08B1 CD1507       CALL ZDV2B
08B4 CDFD07       CALL ZAB2B            ;ABS(DE*172/100)
08B7 F5           PUSH AF              ;SAVE CARRY FLAG
08B8 21D308       LD   HL,SIND3          ;ABS(DE*172/100)+5
08BB CD9F06       CALL ZAD2B
08BE 010000       LD   BC,0000          ;CLEAR MSB
08C1 21D508       LD   HL,SIND4          ;(ABS(DE*172/100)+5)/10
08C4 CD1507       CALL ZDV2B

```

```

08C7 F1          POP  AF          ;LOAD CARRY FLAG
08C8 3003        JR    NC,SIN1      ;IF CARRY=0 THEN RETURN
08CA CDF507      CALL ZSN2B      ;IF CARRY=1 THEN SIGN CHANGE
08CD E1          SIN1: POP  HL
08CE C9          RET

;
08CF AC00        SIND1: DW  172
08D1 6400        SIND2: DW  100
08D3 0500        SIND3: DW   5
08D5 0A00        SIND4: DW  10

;
;COMPARE with SIGN
;
;(IX+1,IX)-(IY+1,IY) IF (IX+1,IX)=(IY+1,IY) THEN ZF=1
;                    (IX+1,IX)<(IY+1,IY) THEN CY=1
;
;                    DE=ABS((IX+1,IX)-(IY+1,IY))
;
;
08D7 DOE5        COMSN: PUSH IX          ;HL=IX
08D9 E1          POP  HL
08DA CD9506      CALL ZLD2B          ;DE=(HL+1,HL)
08DD CB7A        BIT  7,D          ;IF DE>=0 THEN A=0 ELSE A0=1
08DF 3E00        LD   A,0
08E1 2802        JR   Z,COMSN1
08E3 CBC7        SET  0,A
08E5 211E30      COMSN1:LD  HL,CPDAT1      ;(CPDAT1+1,CPDAT1)=(IX+1,IX)
08E8 CD9A06      CALL ZST2B
08EB FDE5        PUSH IY          ;HL=IY
08ED E1          POP  HL
08EE CD9506      CALL ZLD2B
08F1 CB7A        BIT  7,D          ;IF DE>=0 THEN ELSE A1=1
08F3 2802        JR   Z,COMSN2
08F5 CBCF        SET  1,A
08F7 212030      COMSN2:LD  HL,CPDAT2
08FA CD9A06      CALL ZST2B

;
08FD FE00        CP   00H          ;+,+
08FF 2007        JR   NZ,COMSN3
0901 CD2509      CALL CPSUB1
0904 CD3209      CALL CPSUB2
0907 C9          RET

;
0908 FE03        COMSN3:CP   3H          ;-,-
090A 2007        JR   NZ,COMSN4
090C CD2509      CALL CPSUB1
090F CD3209      CALL CPSUB2
0912 C9          RET

;
0913 FE01        COMSN4:CP   1H          ;-,+
0915 2007        JR   NZ,COMSN5
0917 FE00        CP   0H          ;CLEAR Z FLAG
0919 37          SCF              ;SET CARRY
091A CD3209      CALL CPSUB2
091D C9          RET

;
091E 1E00        COMSN5:LD  E,0          ;+,-
0920 AB          XOR  E          ;CLEAR CARRY & Z FLAG
0921 CD3209      CALL CPSUB2

```

```

0924 C9          RET
;
0925 211E30     CPSUB1:LD   HL,CPDAT1
0928 CD9506     CALL  ZLD2B
092B 212030     LD     HL,CPDAT2
092E CDB506     CALL  ZCP2B
0931 C9          RET
;
0932 F5          CPSUB2:PUSH AF
0933 211E30     LD     HL,CPDAT1
0936 CD9506     CALL  ZLD2B
0939 212030     LD     HL,CPDAT2
093C CDAA06     CALL  ZSB2B
093F CDFD07     CALL  ZAB2B
0942 F1          POP   AF
0943 C9          RET
;
;SEARCH LIMIT VALUE
;
0944 DD210712  SLV:  LD     IX,XYIBUF+7      ;IX=Y2 ADDRESS
0948 FD210312  LD     IY,XYIBUF+3      ;IY=Y1 ADDRESS
094C 210014     LD     HL,SAIADR        ;SET SLV TOP ADDRESS TO SAIDAT
094F FDE5       PUSH  IY          ;SAVE TOP DATA AS SAITAN DATA
0951 D1         POP   DE
0952 CD9A06     CALL  ZST2B
0955 23         INC  HL
0956 23         INC  HL
0957 222630     LD     (SAIDAT),HL
095A 3A0012     LD     A,(XYIBUF)      ;B=DATA NUMBER
095D FE01       CP     1
095F CA240A     JP     Z,SLVLP3
0962 FE02       CP     2
0964 CA1F0A     JP     Z,SLVLP2
0967 47         LD     B,A
0968 05         DEC  B          ;SET LOOP COUNTER (B-2)
0969 05         DEC  B
096A CDD708     CALL  COMSN
096D 3E00       LD     A,0
096F 2004       JR     NZ,SLV1
0971 CBE7       SET  4,A          ;Y2=Y1
0973 1804       JR     SLV2
0975 3002     SLV1: JR     NC,SLV2
0977 CBC7       SET  0,A          ;Y2<Y1
0979 212230     SLV2: LD     HL,DEFDA1      ;DEFDA1=IY2-Y1I
097C CD9A06     CALL  ZST2B
097F 322830     LD     (YNDAT),A      ;(YNDAT)=Y1,Y2
0982 FD23       INC  IY          ;COUNT UP YN+1 ADDRESS
0984 FD23       INC  IY
0986 FD23       INC  IY
0988 FD23       INC  IY
098A FD23     SLVLP: INC  IY
098C FD23       INC  IY
098E FD23       INC  IY
0990 FD23       INC  IY
0992 CDD708     CALL  COMSN
0995 3A2830     LD     A,(YNDAT)      ;A=YN-1,YN
0998 0E00       LD     C,0
099A 2004       JR     NZ,SLV3

```

```

099C CBE9          SET 5,C          ;YN=YN+1
099E 1804          JR SLV4
09A0 3002      SLV3: JR NC,SLV4
09A2 CBC9          SET 1,C          ;YN<YN+1
09A4 F5          SLV4: PUSH AF
09A5 3E00          LD A,0
09A7 CB69          BIT 5,C          ;IF YN=(YN+1) THEN YN=(YN+1)
09A9 2804          JR Z,SL1
09AB CBE7          SET 4,A          ;b5--->b4
09AD 180A          JR SL3
09AF CB49      SL1:  BIT 1,C          ;IF YN<(YN+1) THEN YN>(YN+1)
09B1 2004          JR NZ,SL2      ;IF YN>(YN+1) THEN YN<(YN+1)
09B3 CBC7          SET 0,A          ;b1--->b0
09B5 1802          JR SL3
09B7 CB87      SL2:  RES 0,A
09B9 322830     SL3:  LD (YNDAT),A
09BC F1           POP AF
09BD B1           OR C
09BE 212430      LD HL,DEFDA2      ;DEFDA2=(YN-(YN+1))
09C1 CD9A06      CALL ZST2B
09C4 FE03        CP 3H          ;YN<YN-1,YN<YN+1
09C6 2810        JR Z,SLV5
09C8 FE12        CP 12H         ;YN=YN-1,YN<YN+1
09CA 2820        JR Z,SLV7
09CC FE00        CP 0H          ;YN>YN-1,YN>YN+1
09CE 2808        JR Z,SLV5
09D0 FE20        CP 20H         ;YN>YN-1,YN=YN+1
09D2 2802        JR Z,SLV10
09D4 1835        JR SLVLP1
09D6 0EFF      SLV10: LD C,0FFH
09D8 212230     SLV5: LD HL,DEFDA1      ;|YN-(YN-1)|>(KCPDA)
09DB CD9506      CALL ZLD2B
09DE 21350A      LD HL,KCPDA
09E1 CDB506      CALL ZCP2B
09E4 3002        JR NC,SLV6
09E6 1823        JR SLVLP1
09E8 2002      SLV6: JR NZ,SLV7
09EA 181F      SLV7: JR SLVLP1
09EC 0C          SLV7: INC C
09ED 2814        JR Z,SLV9
09EF 212430      LD HL,DEFDA2      ;|YN-(YN+1)|>(KCPDA)
09F2 CD9506      CALL ZLD2B
09F5 21350A      LD HL,KCPDA
09F8 CDB506      CALL ZCP2B
09FB 3002        JR NC,SLV8
09FD 180C        JR SLVLP1
09FF 2002      SLV8: JR NZ,SLV9
0A01 1808        JR SLVLP1
0A03 CD2B0A     SLV9: CALL SLVSUB
0A06 23          INC HL
0A07 23          INC HL
0A08 222630      LD (SAIDAT),HL
0A08 FDE5      SLVLP1: PUSH IY      ;IX=YN+1 ADDRESS
0A0D DDE1        POP IX
0A0F 212430      LD HL,DEFDA2      ;(DEFDA2)=(DEFDA1)
0A12 CD9506      CALL ZLD2B
0A15 212230      LD HL,DEFDA1
0A18 CD9A06      CALL ZST2B
0A1B 05          DEC B
0A1C C28A09      JP NZ,SLVLP

```

```

0A1F CD2B0A  SLVLP2:CALL SLVSUB
0A22 23      INC HL
0A23 23      INC HL
0A24 11FFFF  SLVLP3:LD DE,0FFFFH      ;SET TERMINATER
0A27 CD9A06  CALL ZST2B
0A2A C9      RET

;
0A2B 2A2630  SLVSUB:LD HL,(SAIDAT)      ;SAVE SLV ADDRESS TO SAIDAT
0A2E DDE5    PUSH IX
0A30 D1      POP DE
0A31 CD9A06  CALL ZST2B
0A34 C9      RET

;
;
0A35 C800    KCPOA: DW 200
;
;
;CALCULATE OBJICTIVE TIME & ANGLE
;
;
0A37 DD210016 COTN: LD IX,ATOBJD
0A38 210000    LD HL,0
0A3E 222430    LD (OBTHI2),HL
0A41 210014    LD HL,SAIADR
0A44 222A30    LD (YOBA1),HL
0A47 210214    LD HL,SAIADR+2
0A4A 222C30    LD (YOBA2),HL

;
0A4D FD2A2C30 COTN1: LD IY,(YOBA2)
0A51 FD6E00    LD L,(IY)
0A54 FD6601    LD H,(IY+1)
0A57 3EFF     LD A,OFFH
0A59 BC       CP H
0A5A 2802     JR Z,COTN2
0A5C 1803     JR COTN3
0A5E BD       COTN2: CP L
0A5F 2819     JR Z,CEND

;
0A61 CD9506    COTN3: CALL ZLD2B
;
;
0A64 FD2A2A30 LD IY,(YOBA1)
0A68 FD6E00    LD L,(IY)
0A6B FD6601    LD H,(IY+1)
0A6E 3EFF     LD A,OFFH
0A70 BC       CP H
0A71 2802     JR Z,COTN4
0A73 180F     JR COTN5
0A75 BD       COTN4: CP L
0A76 2802     JR Z,CEND
0A78 180A     JR COTN5

;
0A7A 11FFFF  CEND: LD DE,0FFFFH
0A7D DDE5    PUSH IX
0A7F E1      POP HL
0A80 CD9A06  CALL ZST2B
0A83 C9      RET

;
0A84 CDAA06  COTN5: CALL ZSB2B      ;DEFYOB=YN-(YN-1)
0A87 212E30  LD HL,DEFYOB
0A8A CD9A06  CALL ZST2B
    
```

```

;
0A8D FD2A2C30      LD   IY,(Y0BA2)
0A91 FD6E00        LD   L,(IY)
0A94 FD6601        LD   H,(IY+1)
0A97 2B            DEC  HL
0A98 2B            DEC  HL
0A99 CD9506        CALL ZLD2B
0A9C FD2A2A30      LD   IY,(Y0BA1)
0AA0 FD6E00        LD   L,(IY)
0AA3 FD6601        LD   H,(IY+1)
0AA6 2B            DEC  HL
0AA7 2B            DEC  HL
0AA8 CDAA06        CALL ZSB2B          ;DEFXOB=XN-(XN-1)
0AAB 213030        LD   HL,DEFXOB
0AAE CD9A06        CALL ZST2B
;
0AB1 2A2A30        LD   HL,(Y0BA1)      ;COUNT UP NEXT DATA ADDRESS
0AB4 23            INC  HL
0AB5 23            INC  HL
0AB6 222A30        LD   (Y0BA1),HL
0AB9 2A2C30        LD   HL,(Y0BA2)
0ABC 23            INC  HL
0ABD 23            INC  HL
0ABE 222C30        LD   (Y0BA2),HL
;
0AC1 212E30        LD   HL,DEFYOB      ;ATN(YN-(YN-1)/XN-(XN-1))
0AC4 CD9506        CALL ZLD2B
0AC7 215008        LD   HL,DVDAT
0ACA CDCF06        CALL ZML2B
0ACD 213030        LD   HL,DEFXOB
0ADD CD1507        CALL ZDV2B
0AD3 CB7A          BIT  7,D            ;IF DE<0 THEN BC=FFFF
0AD5 2805          JR   Z,COTN6
0AD7 01FFFF        LD   BC,0FFFFH
0ADA 1803          JR   COTN7
0ADC 010000        COTN6: LD  BC,0
0ADF 214E08        COTN7: LD  HL,TANDAT
0AE2 CD1507        CALL ZDV2B
0AE5 213230        LD   HL,OBTHI1
0AE8 CD9A06        CALL ZST2B
0AEB 212430        LD   HL,OBTHI2
0AEE CDAA06        CALL ZSB2B
0AF1 00E5          PUSH IX
0AF3 E1            POP  HL
0AF4 C0BB07        CALL ZBD2B
0AF7 213230        LD   HL,OBTHI1
0AFA 112430        LD   DE,OBTHI2
0AFD 010200        LD   BC,2
0B00 EDB0          LDIR
0B02 DD23          INC  IX
0B04 DD23          INC  IX
;
0B06 213030        LD   HL,DEFXOB      ;(XN-(XN-1))*2
0B09 CD3A0B        CALL COTSUB
0B0C 213630        LD   HL,XSQUR
0B0F CD9A06        CALL ZST2B
;
0B12 212E30        LD   HL,DEFYOB      ;(YN-(YN-1))*2
0B15 CD3A0B        CALL COTSUB
0B18 213830        LD   HL,YSQUR

```

```

081B CD9A06          CALL ZST2B
;
081E 213630          LD   HL, XSQR          ;(XN-(XN-1))*2+(YN-(YN-1))*2
0821 CD9506          CALL ZLD2B
0824 213830          LD   HL, YSQR
0827 CD9F06          CALL ZAD2B
;
082A CD6207          CALL ZSQ2B          ;SQR((XN-(XN-1))*2+(YN-(YN-1))*2)
;
082D DDE5            PUSH IX
082F E1              POP  HL
0830 CDBB07          CALL ZBD2B
0833 DD23            INC  IX
0835 DD23            INC  IX
0837 C34D0A          JP   COTN1
;
083A CD9506          COTSUB:CALL ZLD2B          ;DATA*DATA/10000
083D ED533A30        LD   (SQR),DE
0841 213A30          LD   HL, SQR
0844 CDCF06          CALL ZML2B
0847 21500B          LD   HL, DV DAT
084A CD1507          CALL ZDV2B
084D C9              RET
;
084E 1200            TANDAT:DW 18
0850 1027            DV DAT: DW 10000
;
0852                END
    
```



```

10 '*****
20 '*
30 '* Simulation Program *
40 '* *
50 '* *
60 '* for Turn Experiment *
70 '* *
80 '* *
90 '*****
100 '
110 WIDTH 80,25:CONSOLE 0,25,0,0:SCREEN 3,0
120 RANDOMIZE:CLS 3
130 PI=3.14159
140 '
150 'Simulate oscilation of combine
160 W1=2*PI*43
170 W2=2*PI*21.5
180 W3=2*PI*6.3
190 W4=2*PI*28.8
200 W5=2*PI*14.6
210 W6=2*PI*43.5
220 W7=2*PI*65.4
230 W8=2*PI*19
240 W9=2*PI/1800:'30 min period
250 SAMPLE.TIME=.05 :'Sampling time = .05 sec
260 GOTO *MAIN
270 '
280 'Simulate Angular Rate Sensor
290 '
300 'Input parameter THETA.DOT (deg/sec)
310 'Output parameter V.ARS (Volt)
320 '
330 *ANGULAR.RATE.SENSOR
340 IF THETA.DOT=0 THEN RETURN
350 IF FLAG=0 THEN TSR=TS+RND*100 ELSE TSR=ST+TSR
360 IF A$<>"Y" OR A$<>"y" THEN 430
370 O1=.133*SIN(W1*TSR)
380 O2=.064*SIN(W2*TSR)
390 O3=.063*SIN(W3*TSR)
400 O4=.045*SIN(W4*TSR)
410 O5=.039*SIN(W5*TSR)
420 O=O1+O2+O3+O4+O5:GOTO 470
430 O1=.075*SIN(W6*TSR)
440 O2=.039*SIN(W7*TSR)
450 O3=.035*SIN(W8*TSR)
460 O=O1+O2+O3
470 V.OF=.005*10*SIN(W9*TSR)
480 V.IN=THETA.DOT/10:'Convert angular rate to volt
490 V.ARS=V.IN+O+V.OF
500 'Check scall over
510 IF V.ARS>12 THEN V.ARS=12
520 IF V.ARS<-12 THEN V.ARS=-12
530 FLAG=1
540 RETURN
550 '
560 'A/D Conversion & Calculate Turn Angle
570 '
580 *ADCONV
590 V.ADC=(INT(V.ARS*1000+.5)/5)*5/1000 :'5mV/digit
600 'Calculate Turn Angle between sampling time
610 ANG=INT(100*V.ADC*SAMPLE.TIME+.5)/10

```

```

620 S. ANG=S. ANG+ANG
630 IF S. ANG>180 THEN S. ANG=S. ANG-360
640 IF S. ANG<-180 THEN S. ANG=S. ANG+360
650 RETURN
660 '
670 '
680 '
690 *MAIN
700 INPUT "Turning Angle degree Right Turn=(-)";C.ANG
710 IF C.ANG<0 THEN XO=250:YO=100 ELSE XO=250:YO=250
720 INPUT "Drive Cutting & Threshing Device (y/n)";As
730 '
740 'Set Initial Value
750 FOR LOOP=1 TO 6
760 CLS 3
770 ST=.01 : 'Step Time (s)
780 RR=105:RL=110 : 'Turn Radius (cm)
790 P=0:Q=0 : 'Central Divider Position
800 A=-170:B=-120 : 'Right Turn center Position
810 C=-170:D=95 : 'Left Turn center Position
820 E=-170:F=0 : '
830 '
840 '
850 'Delay Time (1/100 s)
860 '
870 DT.R1=7 : 'Right Sol ON
880 DT.R2=14 : ' OFF
890 DT.L1=7 : 'Left Sol ON
900 DT.L2=12 : ' OFF
910 DT.UP=52 : 'Cutter Down -> Up
920 DT.DN=74 : ' Up -> Down
930 DT.N1=71 : 'Neutral -> 1th
940 DT.1N=6 : '1th -> Neutral
950 DT.NR=48 : 'Neutral -> Reverse
960 DT.RN=84 : 'Reverse -> Neutral
970 '
980 '
990 READ V
1000 BETA=-V*ST/RR : 'Right Turn Angle (rad)
1010 ALFA=V*ST/RL : 'Left Turn Angle (rad)
1020 LS=V*ST/(SQR((P-E)^2+(Q-F)^2))
1030 '
1040 DATA 28,35,39,51,64,69 : 'Velocity Data
1050 '
1060 'Turn Experiment
1070 '
1080 'Simulation Start
1090 '
1100 TS=0 : 'Start Time=0
1110 FLAG=0
1120 FOR L=1 TO 200 : '2s FORWARD
1130 GOSUB *STRAIGHT
1140 GOSUB *PLOT
1150 TS=TS+ST : 'Time Count
1160 NEXT
1170 RESET.TIME=RND/5 : 'SUBCPU Reset
1180 FOR L=1 TO RESET.TIME*100+DT.L1 : 'Delay time
1190 GOSUB *STRAIGHT
1200 GOSUB *PLOT
1210 TS=TS+ST
1220 NEXT
1230 IF C.ANG<0 THEN R=RR ELSE R=RL

```

```

1240 S. ANG=0: THETA. DOT=V*180/PI/R
1250 IF C. ANG<0 THEN THETA. DOT=-THETA. DOT
1260 AT%=1: N%=INT(SAMPLE. TIME/ST)
1270 GOSUB *ANGULAR. RATE. SENSOR
1280 IF AT%=(AT%*N%)*N% THEN GOSUB *ADCONV ELSE 1310
1290 IF C. ANG>0 THEN IF S. ANG>C. ANG THEN 1350
1300 IF C. ANG<0 THEN IF S. ANG<C. ANG THEN 1350
1310 IF C. ANG>0 THEN GOSUB *LEFT ELSE GOSUB *RIGHT
1320 GOSUB *PLOT
1330 TS=TS+ST: AT%=AT%+1
1340 GOTO 1270
1350 FOR L=1 TO DT. L2
1360 IF C. ANG>0 THEN GOSUB *LEFT ELSE GOSUB *RIGHT
1370 GOSUB *PLOT
1380 TS=TS+ST
1390 NEXT
1400 FOR L=1 TO 200
1410 GOSUB *STRAIGHT
1420 GOSUB *PLOT
1430 TS=TS+ST
1440 NEXT
1450 TAU=ATN((Q-F)/(P-E))
1460 LPRINT "TRAVELING SPEED      ";V/100;"m/s"
1470 LPRINT "SETTING TURN ANGLE    ";C. ANG;"°"
1480 LPRINT "SUBCPU TURN ANGLE      ";S. ANG;"°"
1490 LPRINT "REAL TURN ANGLE        ";TAU*180/PI;"°"
1500 IF A$="Y" OR A$="y" THEN LPRINT "DRIVE CUTTER & THRESHER"
      ELSE LPRINT "ONLY RUNNING"
1510 LPRINT:LPRINT:LPRINT
1520 INPUT "Hard Copy (y/n)";H$
1530 IF H$="y" OR H$="Y" THEN CLS :COPY
1540 NEXT
1550 END
1560 '
1570 'Calculate Traveling Direction
1580 '
1590 ' Display Tracking
1600 *PLOT
1610 PSET(P+XO, YO-Q), 5: PSET(E+XO, YO-F), 4
1620 PSET(A+XO, YO-B), 6: PSET(C+XO, YO-D), 2
1630 RETURN
1640 '
1650 'Right turn  beta<=0
1660 '
1670 *RIGHT
1680 IF BACK=1 THEN BETA=-BETA
1690 CS=COS(BETA): SN=SIN(BETA)
1700 P=(P-A)*CS-(Q-B)*SN+A
1710 Q=(P-A)*SN+(Q-B)*CS+B
1720 C=(C-A)*CS-(D-B)*SN+A
1730 D=(C-A)*SN+(D-B)*CS+B
1740 E=(E-A)*CS-(F-B)*SN+A
1750 F=(E-A)*SN+(F-B)*CS+B
1760 IF BACK=1 THEN BETA=-BETA
1770 BACK=0
1780 RETURN
1790 '
1800 'Left turn  alfa>=0
1810 '
1820 *LEFT
1830 CS=COS(ALFA): SN=SIN(ALFA)
1840 P=(P-C)*CS-(Q-D)*SN+C

```

```
1850 Q=(P-C)*SN+(Q-D)*CS+D
1860 A=(A-C)*CS-(B-D)*SN+C
1870 B=(A-C)*SN+(B-D)*CS+D
1880 E=(E-C)*CS-(F-D)*SN+C
1890 F=(E-C)*SN+(F-D)*CS+D
1900 RETURN
1910 '
1920 'Straight
1930 '
1940 *STRAIGHT
1950 XS=(P-E)*LS
1960 YS=(Q-F)*LS
1970 IF BACK=1 THEN XS=-XS:YS=-YS
1980 P=P+XS:Q=Q+YS
1990 A=A+XS:B=B+YS
2000 C=C+XS:D=D+YS
2010 E=E+XS:F=F+YS
2020 BACK=0
2030 RETURN
```